Optimizing LLMs for Italian: Reducing Token Fertility and Enhancing Efficiency Through Vocabulary Adaptation

Luca Moroni[1]*, Giovanni Puccetti[2]*, Pere-Lluis Huguet Cabot[1], Andrei Stefan Bejgu[4]
Edoardo Barba[1], Alessio Miaschi[3]
Felice Dell'Orletta[3], Andrea Esuli[2], Roberto Navigli[1]
[1]Sapienza University of Rome `{surname}@diag.uniroma1.it`
[2]ISTI-CNR `{name.surname}@isti.cnr.it`
[3]ILC-CNR `{name.surname}@ilc.cnr.it`
[4]Babelscape `{surname}@babelscape.com`

Abstract

The number of pretrained Large Language Models (LLMs) is increasing steadily, though the majority are designed predominantly for the English language. While state-of-the-art LLMs can handle other languages, due to language contamination or some degree of multilingual pretraining data, they are not optimized for non-English languages, leading to inefficient encoding (high token "fertility") and slower inference speed. In this work, we thoroughly compare a variety of vocabulary adaptation techniques for optimizing English LLMs for the Italian language, and put forward Semantic Alignment Vocabulary Adaptation (SAVA), a novel method that leverages neural mapping for vocabulary substitution. SAVA achieves competitive performance across multiple downstream tasks, enhancing grounded alignment strategies. We adapt two LLMs: *Mistral-7B-v0.1*, reducing token fertility by 25%, and *Llama-3.1-8B*, optimizing the vocabulary and reducing the number of parameters by 1 billion. We show that, following the adaptation of the vocabulary, these models can recover their performance with a relatively limited stage of continual training on the target language. Finally, we test the capabilities of the adapted models on various multi-choice and generative tasks.[1]

1 Introduction

Large Language Models (LLMs) have gained immense popularity and are increasingly being utilized across a wide range of applications (Radford et al., 2019; Kojima et al., 2022). Despite their impressive performance, these models are mainly English-centric, that is, most state-of-the-art models are designed and pre-trained on datasets with a primary focus on English (Jiang et al., 2023; Dubey et al., 2024; Mesnard et al., 2024). Although native multilingual models – i.e. fully pre-trained in multiple target languages – have been released over the years (Le Scao et al., 2023), they still fall short of achieving performance levels comparable to models pre-trained in English. The primary challenge is addressing underrepresented languages, where large, clean, open-access text corpora are often scarce (Weber et al., 2024; Nguyen et al., 2024). This scarcity is problematic because models require vast amounts of high-quality data to achieve satisfactory performance (Hoffmann et al., 2022). Moreover, multilingual models generally reach suboptimal performance due to the well-known *curse of multilinguality* (Conneau et al., 2020).

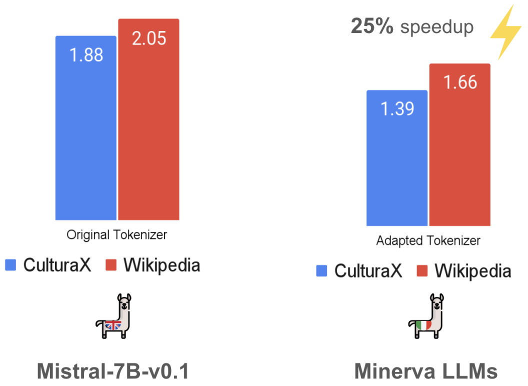

Figure 1: Fertility for two different tokenizers, *Mistral-7B-v0.1* (left) and *Minerva* (right), over Italian texts from CulturaX (blue) and Wikipedia (red).

A promising solution to these challenges is the adaptation of pretrained English LLMs to other languages (Chau et al., 2020). Recent studies highlight that fine-tuning English-centric models to support other languages yields substantial benefits, allowing for efficient adaptation while minimizing computational resources and training time. This method reduces both the training budget and the number of tokens required, demonstrating competitive performance even in low-resource scenarios (Koto et al., 2021; Minixhofer et al., 2022; Gee et al., 2022; Ostendorff and Rehm, 2023).

* Those authors contributed equally.

[1] We release our code and models at `https://github.com/SapienzaNLP/sava`

Findings of the Association for Computational Linguistics: NAACL 2025, pages 6661–6675
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

Another important aspect, alongside the downstream performance of language models, is the *tokenizer's fertility* in target languages. LLMs rely on a tokenizer, which is trained on a mix of text (either LLM's training data or not), transforming raw text into word-piece tokens; *fertility* is the average number of tokens in which a word is split (Brown et al., 1993). The fertility of a tokenizer is highly sensitive to the language and the type of text it was trained on, as well as the text on which the fertility itself is measured. Figure 1 shows an example of this phenomenon comparing the fertility of *Minerva-LLMs*, a family of Italian-first LLMs (Orlando et al., 2024), and *Mistral-7B-v0.1*, an English-first LLM (Jiang et al., 2023), on two corpora in Italian.

In this work, we explore the adaptation of two state-of-the-art English LLMs to the Italian language using both vocabulary adaptation and continual learning. Additionally, we introduce a novel vocabulary adaptation technique called *Semantic Alignment Vocabulary Adaptation (SAVA)* and conduct a comprehensive comparison with recent approaches (Gee et al., 2022; Ostendorff and Rehm, 2023), examining the impact of vocabulary substitution on model performance throughout the adaptation process. After the adaptation of the vocabulary, when tokenizing Italian texts, we are able to reduce the fertility of *Mistral-7B-v0.1* by 25% and of *Llama-3.1-8B* by 16%. As regards *Mistral-7B-v0.1*, we do not increase its vocabulary size or model parameters, while for *Llama-3.1-8B*, we effectively reduce its vocabulary size by 75% thereby reducing the final model size by 10%. Overall, we reduce memory and compute footprint of the models. To summarize, the main contributions are:

- Introducing an effective approach for adapting tokenizers and vocabularies of generative models, leading to competitive performance over existing methods across several downstream benchmarks;

- Providing a detailed comparative analysis of various tokenizer adaptation techniques, with a focus on continual training in low- to mid-resource scenarios.

- Analyzing the embedding representations learned through different adaptation techniques, offering a deeper understanding of how vocabulary modifications impact model performance and generalization.

2 Related Work

Language-Adaptive Pretraining Designing LLMs in a target language and thus training them from scratch is the best approach to obtain an adequate token fertility from the outset and minimize interference from pretrained data on different languages. However, this approach is often impractical, especially in low-resource settings and on a low computational budget. For this reason, several recent studies (de Vries and Nissim, 2021; Gee et al., 2022; Csaki et al., 2024) have focused on the adaptation of pretrained LLMs to new languages. Pretrained LLMs can be adapted to a specific language using a small quantity data compared to what is needed in the pretraining stage. A straightforward approach to achieving this is Language-Adaptive Pre-Training (LAPT), utilized by Chau et al. (2020) in a multilingual setting where they tested continual training of multilingual LLMs on target languages. Interestingly, LAPT was previously proposed on encoder-only architectures by Gururangan et al. (2020), where they successfully adapted RoBERTa (Zhuang et al., 2021) models in a biomedical domain. In LAPT, models do not undergo any structural change to their architecture. This usually results in performance improvements, however it does not address the limitations of using a sub-optimal tokenizer that is less suited to the encoding of different languages. Regarding LAPT research in English-to-Italian models, there have been several attempts, most notably *LLaMAntino-2-LLMs*, which is a fine-tuning of LLama 2 on Italian translated conversations (Basile et al., 2023), and *LLaMAntino-3-ANITA-8B-Inst-DPO-ITA*, a more recent effort that is built upon *Llama-3-8B* using a similar approach (Polignano et al., 2024).

Vocabulary Adaptation Techniques To tackle the fertility issue, recent research has focused on improving language adaptation by modifying the tokenizer and vocabulary of pretrained LLMs to better fit the target language. Several efforts in this area have shown the effectiveness of vocabulary adaptation techniques. Minixhofer et al. (2022) and Liu et al. (2024) propose to replace the tokenizer of a pretrained LLM, along with its corresponding embedding layer, relying on a bilingual dictionary-based, or graph-based, token mapping. Generally, the main difference between various vocabulary adaptation techniques lies in how the embedding space of the respective model is initialized during adaptation. More effort was made by Ostendorff and Rehm (2023); Dobler and de Melo (2023) who use the embeddings from a helper model trained alongside the desired tokenizer. They utilize geometrical similarities in the helper model's embedding structure to initialize the tokens' representations of the target model effectively. In parallel, Gee et al. (2022) proposed a simple heuristic, initializing target vocabulary tokens as the average of their corresponding sub-tokens in the source vocabulary. Another study by Koto et al. (2021) put forward an adaptation technique, they rely on `FastText`[2] embedding space to learn a linear mapping, to perform vocabulary adaptation of BERT-based models.

Unlike previous studies, we thoroughly analyze existing adaptation heuristics, focusing on decoder-only generative models adapted to Italian. We present a novel heuristic that utilizes a helper embedding space, optimized for the target language, to map and initialize target vocabulary tokens.

3 Methodology

In this section, we formalize the methodologies used to adapt pretrained LLMs to a target language. The following subsections outline the techniques employed to modify the vocabulary of pretrained LLMs and describe the process of adapting them to a target language. Finally, we describe the last step of the adaptation, that is, the continual training step.

3.1 Vocabulary Adaptation

All the vocabulary adaptation methodologies share a similar objective: substituting the tokenizer and its vocabulary, and replacing the model embeddings (both embedding module and language model head) with one more suited for the target language.

In our setting, we have a source pretrained LLM, M_s, with its embedding matrix E_s[3], tokenizer T_s, and vocabulary V_s. To adapt our model to a target language, we have a target tokenizer T_t and vocabulary V_t suited to encoding texts in the target language, which we want to make M_s compatible with. In some cases, we also have access to a helper model, M_h, which is an LLM, usually smaller than M_s, whose embeddings are noted with E_h. The helper model is trained using T_t and V_t. We use the superscript notation E^{t_i} to indicate the representation of the token t_i on the matrix embedding E.

The objective is to adapt the source model embeddings E_s, which all these methods operate on, so as to obtain E_t based on the target tokenizer T_t and the target vocabulary V_t.

First, the target embeddings are initialized by keeping the same representation from E_s for the tokens in the intersection of both vocabularies, while a function g is applied to the remaining ones:

$$E_t^{t_i} = \begin{cases} g(t_i, \cdot), & t_i \in V_t \setminus V_s \\ E_s^{t_i}, & t_i \in V_s \cap V_t \end{cases}$$

The difference between these methods lies in g, used to initialize the tokens that are in V_t and not in V_s. This function has access to the source embeddings E_s, vocabulary V_s and tokenizer T_s and possibly the embeddings, vocabulary, and tokenizer of a helper model E_h, V_h and T_h, respectively.

Therefore, each method is defined by its respective g function, as detailed below.

Random As a baseline approach, we initialize the tokens outside the intersection with a random representation, given by a normal distribution, with the mean and the variance defined by the source embedding space:

$$g_{random}(t_i, E_s) = \mathcal{N}(\mu(E_s), \sigma^2(E_s))$$

FVT Gee et al. (2022) introduced Fast Vocabulary Transfer (FVT) for vocabulary adaptation, which consists of an efficient way to initialize the intersection tokens in the target embedding space. Here, each target token is computed with the average of the embedding source tokens given by the source tokenizer, i.e. the resulting tokens when we tokenize the target token t_i with T_s:

$$g_{fvt}(t_i, E_s, T_s) = \frac{1}{|T_s(t_i)|} \cdot \sum_{t_j \in T_s(t_i)} E_s^{t_j}.$$

CLP Ostendorff and Rehm (2023) and in parallel Dobler and de Melo (2023) introduced a heuristic to initialize out-of-inventory tokens relying on the space structure of the helper embedding space. Both approaches compute similarity scores between the tokens in $V_t \setminus V_s$ against the ones in $V_t \cap V_s$, on the embedding space of the helper

[2] https://fasttext.cc/

[3] Here, we assume tied-weights, i.e., shared embedding module and language model head. When this is not the case, the approach is symmetric, as if there were two embedding matrices.

model E_h. Such similarities are used to construct a representation of the out-of-inventory tokens in the target embedding matrix E_t, relying on the source embedding E_s representations:

$$g_{clp}(t_i, E_s, V_s, E_h, V_h) = \sum_{t_j \in V_t \cap V_s} E_s^{t_j} \cdot \alpha(E_h^{t_i}, E_h^{t_j})$$

where $\alpha(\cdot, \cdot)$ indicates a similarity score between two tokens in E_h. Here, we rely on the similarity function used by Ostendorff and Rehm (2023) computed as normalized cosine similarity.

SAVA Mapping embedding representations between the embedding spaces of two different models using a linear model comes with theoretical justification. Moschella et al. (2023) and Maiorca et al. (2024) have shown that the embeddings of different models are related by a *conformal translation*, or more generally, by a *linear mapping* between such spaces. Inspired by the findings of Maiorca et al. (2024) and by the intriguing effort of Koto et al. (2021), we propose a technique to perform vocabulary adaptation for generative models called *Semantic Alignment Vocabulary Adaptation (SAVA)*. In our approach, we rely on a helper model embedding E_h from an LLM and learn a linear mapping ϕ between $E_h \subseteq \mathbb{R}^m$ and $E_s \subseteq \mathbb{R}^n$. We train a single-layer Feed Forward Network (FFN) to map the helper embedding space onto the source embedding one:

$$\phi \;:\; x \mapsto y \;\mid x \in \mathbb{R}^m,\; y \in \mathbb{R}^n,$$
$$g_{sava}(t_i, E_h) = \phi(E_h^{t_i}) \qquad (1)$$

The goal in training ϕ is to obtain a mapping between the representations of the tokens of the helper model and those of the source one. To train it, we use the tokens in the intersection $V_s \cap V_t$ since they have a representation according to both the source and the helper model, and we can train a linear map between the representations in E_s and those in E_h. Then, as outlined from equation 1 we use ϕ to map the tokens not present in the source vocabulary ($V_t \setminus V_s$) into the source embedding space. Therefore, our objective is to find:

$$\phi(x) = Wx + b,$$

such that,

$$\min_{W,b} \sum_{t_i \in V_s \cap V_t} \left\| W E_h^{t_i} + b - E_s^{t_i} \right\|^2 .$$

where $W \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^n$ are the parameters of our linear mapping. More technical details about the training of the linear mapping are provided in Appendix A.

3.2 Continual Training

While re-initializing embeddings through vocabulary adaptation techniques enables zero-shot language modeling, the resulting language model often lacks proficiency in the new language. We address this by performing continual training on a mixture of source and target languages, which allows the model to retain performance in the source language while improving in the target language.

To achieve a robust comparison, we adapt pretrained LLMs to the target language using all the vocabulary adaptation heuristics discussed above. We also present results from continual training of the base model on the target language (LAPT). While less disruptive, this approach does not alter the vocabulary or tokenizer, preserving its fertility.

4 Experimental Setup

This section describes the setup of our experiments where we adapt two popular LLMs, specifically *Mistral-7B-v0.1* (Jiang et al., 2023) and *Llama-3.1-8B* (Dubey et al., 2024). In the following subsections we report the settings used to do vocabulary adaptation, continual training and evaluation.

4.1 Vocabulary Adaptation

To adapt English models to the Italian language we rely on the *Minerva-LLMs* model family and its tokenizer (Orlando et al., 2024). The models of the *Minerva-LLMs* family are trained from scratch on an Italian-English dataset, i.e. CulturaX (Nguyen et al., 2024). At the time of writing, three different models have been released, *Minerva-350M*, *Minerva-1B*, and *Minerva-3B*, with the same tokenizer.

The *Minerva-LLMs* tokenizer shares 16,438 tokens with *Mistral-7B-v0.1* and 20,358 tokens with *Llama-3.1-8B*. For both *CLP* and *SAVA*, we use *Minerva-3B* as the helper model.[4] Notably, as shown in Table 1, adapting a large model like *Llama-3.1-8B* with *Minerva-LLMs* tokenizer significantly reduces the vocabulary size (by 75%) and thus results in fewer parameters. The adapted

[4] We conduct ablation studies for the SAVA method, changing the number of tokens used to train ϕ and the size of the helper model. Some considerations are reported in Appendix B.

Model	Num. Tokens	Num. Parameters
Mistral-7B-v0.1	32000	7.24B
Mistral-7B-v0.1 a.w. Minerva	32768	7.25B
LLaMa-3-8B	128256	8.03B
LLaMa-3-8B a.w. Minerva	32768	7.25B

Table 1: Comparisons of model parameter counts and vocabulary size with and without adaptation (a.w. stands for *adapted with*).

	Fertility ↓			
Model	**CX IT**	**CX EN**	**Wp IT**	**Wp EN**
Mistral-7B-v0.1	1.88	1.32	2.05	1.57
Minerva	**1.39**	1.32	**1.66**	1.59
LLaMa-3-8B	1.67	**1.15**	1.80	**1.31**

Table 2: Fertility of different tokenizers on CulturaX (CX) and Wikipedia (Wp).

Llama-3.1-8B has 7.25B parameters compared to the original 8B, resulting in a 10% reduction in model size.

As a further improvement, substituting the *Mistral-7B-v0.1* and *Llama-3.1-8B* tokenizers with *Minerva-LLMs* one has a significant impact on the fertility in the Italian language. As shown in Table 2, the *Minerva-LLMs* tokenizer has on average 25% of fertility gain compared to the *Mistral-7B-v0.1* tokenizer on two Italian text sources, CulturaX (CX) and Wikipedia (Wp). In the same setting, *Llama-3.1-8B* improves its fertility up to 16% on Italian text relying on the *Minerva-LLMs* tokenizer.

4.2 Continual Training

To perform continual training we use CulturaX, a large-scale multilingual dataset that has been successfully used in large-scale continual training experiments on languages spoken within the European Union, including Italian.[5] We aim to compare all methods on a fixed amount of compute budget, i.e. number of tokens. Due to a constrained computational budget, we decide to stop training after a threshold of 12B training tokens.

We subsample training data from the Italian and English splits of CulturaX to create a dataset composed of 75% Italian tokens and 25% English tokens, as proposed by Csaki et al. (2024).

We use packing to fit all the tokens into sequences of a fixed length. The learning rate is fixed for all runs at 10^{-5}.

For *Mistral-7B-v0.1*, training is done on 16 nodes on the Leonardo Supercomputer (each node uses 4 x 64 GB A100) maintaining a global batch size of 3072, and a sequence length of 2048. For *Llama-3.1-8B* we change the sequence length of the training data to 8192, and set the global batch size to 512. When training both models we do not freeze any parameter and let them all update. We perform continual training, allowing the models to process approximately 12 billion tokens. Specifically, we train *Mistral-7B-v0.1* for 2000 batches and *Llama-3.1-8B* for 3000 batches. We use *llm-foundry* for training[6] and for the remaining hyperparameters we use the default settings provided by the library. See Appendix C for an estimation of the CO_2 cost of the experiments carried out in this work.

4.3 Evaluation

To evaluate our models we rely on the LM-Evaluation-Harness library (Gao et al., 2024), for multiple-choice (MC) benchmarks, using the perplexity evaluation method. As MC benchmarks, we use the translated section of ITA-Bench (Moroni et al., 2024), a suite of benchmarks automatically translated from English to Italian.

During continual training we evaluate our models every 200 batches for *Mistral-7B-v0.1* and 300 batches for *Llama-3.1-8B* in a 0-shot scenario; in this way, each subsequent checkpoint is evaluated consistently on the same number of tokens. To assess the reasoning capabilities of the adapted models, we use a variety of benchmarks: **MMLU** (Hendrycks et al., 2021), **BOOLQ** (Clark et al., 2019), **ARC-easy** (Clark et al., 2018), **PIQA** (Bisk et al., 2020), **SciQ** (Welbl et al., 2017), and **Hellaswag** (Zellers et al., 2019).

We also measure the model performance on generative tasks, focusing on two tasks: automatic translation, **FLoRes** benchmark (Costa-jussà et al., 2022), and question answering, **SQuAD-it** (Croce et al., 2018), a version of **SQuAD** (Rajpurkar et al., 2016) automatically translated into Italian. We used `vLLM` (Kwon et al., 2023) as our generation pipeline. More details related to the generation techniques can be found in Appendix D.

5 Results

In this section, we discuss the results obtained from evaluating the adapted models. We begin by examining the scores on multiple-choice benchmarks, followed by a separate analysis of performance on

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

[6] https://github.com/mosaicml/llm-foundry

Model	Hellaswag	MMLU	Arc Easy	PIQA	SCIQ	BOOLQ	AVG
Mistral-7B-v0.1	56.50$_{\pm 0.49}$	47.42$_{\pm 0.42}$	61.67$_{\pm 1.01}$	67.24$_{\pm 1.14}$	84.75$_{\pm 1.16}$	75.01$_{\pm 0.75}$	65.43
200 Training Steps							
Random	55.60$_{\pm 0.49}$	42.48$_{\pm 0.42}$	57.92$_{\pm 1.02}$	68.05$_{\pm 1.16}$	75.46$_{\pm 1.39}$	72.29$_{\pm 0.78}$	61.96
FVT	56.34$_{\pm 0.49}$	**44.28**$_{\pm 0.42}$	60.42$_{\pm 1.01}$	**69.90**$_{\pm 1.14}$	**80.48**$_{\pm 1.28}$	**74.52**$_{\pm 0.76}$	**64.32**
CLP	54.74$_{\pm 0.49}$	42.50$_{\pm 0.42}$	57.62$_{\pm 1.02}$	67.74$_{\pm 1.16}$	76.82$_{\pm 1.36}$	68.07$_{\pm 0.81}$	61.24
SAVA	**56.73**$_{\pm 0.49}$	44.23$_{\pm 0.42}$	**60.90**$_{\pm 1.01}$	69.72$_{\pm 1.14}$	79.22$_{\pm 1.31}$	73.30$_{\pm 0.77}$	64.01
LAPT	58.29$_{\pm 0.49}$	49.31$_{\pm 0.42}$	63.00$_{\pm 1.00}$	69.84$_{\pm 1.14}$	84.13$_{\pm 1.18}$	75.07$_{\pm 0.75}$	66.60
2000 Training Steps							
Random	58.43$_{\pm 0.49}$	46.95$_{\pm 0.42}$	62.87$_{\pm 1.00}$	71.39$_{\pm 1.12}$	81.62$_{\pm 1.25}$	72.47$_{\pm 0.78}$	65.62
FVT	59.00$_{\pm 0.49}$	47.35$_{\pm 0.42}$	**63.52**$_{\pm 0.99}$	**71.51**$_{\pm 1.12}$	**84.55**$_{\pm 1.16}$	75.74$_{\pm 0.74}$	66.94
CLP	59.21$_{\pm 0.49}$	47.10$_{\pm 0.42}$	63.47$_{\pm 0.99}$	70.77$_{\pm 1.13}$	84.44$_{\pm 1.17}$	**76.75**$_{\pm 0.73}$	66.95
SAVA	**59.41**$_{\pm 0.49}$	**47.57**$_{\pm 0.42}$	63.39$_{\pm 0.99}$	71.02$_{\pm 1.12}$	**84.55**$_{\pm 1.16}$	76.02$_{\pm 0.74}$	**66.99**
LAPT	60.51$_{\pm 0.48}$	46.63$_{\pm 0.42}$	64.99$_{\pm 0.99}$	71.21$_{\pm 1.12}$	85.90$_{\pm 1.12}$	76.17$_{\pm 0.74}$	67.56

Table 3: 0-shot results over **Italian** translated benchmarks for *Mistral-7B-v0.1* adapted models.

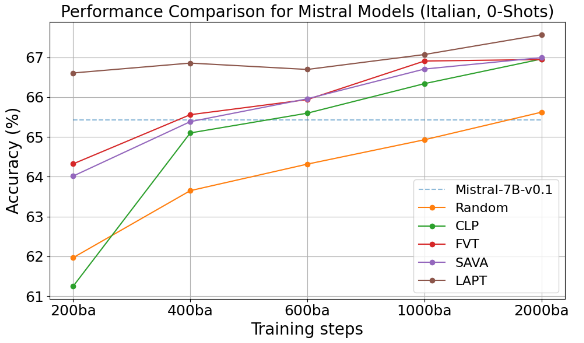

Figure 2: Average performance of *Mistral-7B-v0.1* based models during training on **Italian** translated benchmarks. The average was calculated over six datasets.

generative benchmarks, specifically *FLoRes* and *SQuAD-it*. In this and subsequent sections we indicate the continual training of the base model without vocabulary adaptation by the *LAPT* acronym.

5.1 Multi-choice Setting

5.1.1 Italian Results

We report results on Italian benchmarks for *Mistral-7B-v0.1* after 200 and 2000 batches in Table 3. From the table we can see that the adapted models reach over random-chance performance at the beginning of training (200-step setting), with *FVT* and *SAVA* achieving higher performance compared to other methods (*CLP* and *Random*). All the vocabulary adaptation heuristics perform worse compared to the *LAPT* technique, which is expected since *LAPT* does not apply any disruptive architectural change to the model. Looking at the results at 2000 batches, we can see that all the adapted models surpass the scores of the base model, and the performance gap with *LAPT* becomes low. Even in this setting, *SAVA* and *FVT* perform well, while *Random* lags behind.

In Figure 2, we present the average scores across the six Italian tasks. *SAVA* and *FVT* consistently achieve higher overall scores throughout the training process, with a more pronounced advantage in the early stages. This highlights the influence of the chosen heuristic, particularly immediately after the vocabulary substitution. *SAVA* and *FVT* achieve results at 400 batches that are comparable to those of the *Random* approach at the end of training, thereby reducing total training time by approximately 80%.

In the case of *Llama-3.1-8B*, Table 4 reports the scores of the adapted models, after 300 and 3000 batches. We show that *FVT* and *SAVA* maintain comparable performance, except for *BOOLQ* where *SAVA* showcases better scores, +4%, even in comparison to the *LAPT* setting. Compared to the adapted models, the *Llama-3.1-8B* model remains a strong baseline on Italian tasks. Still in that setting, we further narrow the performance gap with the *LAPT* model using both vocabulary adaptation heuristics. In Figure 3, we report the average scores on Italian tasks and observe a constant improvement through the training steps.

5.1.2 English Results

Including English in the evaluation allows us to assess whether performance on the source language is preserved during continual training, for both *Mistral-7B-v0.1* and *Llama-3.1-8B*. As mentioned in Section 4, we train on mainly Italian data and a smaller portion of English (25% of the total).

Figure 4 reports the average scores during training on English texts for *Mistral-7B-v0.1*. We can see that all trained models reach a comparable av-

Model	Hellaswag	MMLU	Arc Easy	PIQA	SCIQ	BOOLQ	AVG
LLaMa-3.1-8B	$57.97_{\pm0.49}$	$54.28_{\pm0.42}$	$60.46_{\pm1.01}$	$68.54_{\pm1.15}$	$82.77_{\pm1.22}$	$74.52_{\pm0.76}$	66.42
300 Training Steps							
FVT	$\mathbf{55.61}_{\pm0.49}$	$\mathbf{50.24}_{\pm0.42}$	$59.38_{\pm1.01}$	$\mathbf{66.99}_{\pm1.17}$	$80.68_{\pm1.27}$	$70.00_{\pm0.80}$	63.81
SAVA	$55.48_{\pm0.49}$	$49.26_{\pm0.42}$	$\mathbf{59.77}_{\pm1.01}$	$66.62_{\pm1.17}$	$\mathbf{81.31}_{\pm1.26}$	$\mathbf{74.43}_{\pm0.76}$	**64.48**
LAPT	$57.92_{\pm0.49}$	$53.10_{\pm0.42}$	$61.32_{\pm1.01}$	$68.97_{\pm1.15}$	$82.56_{\pm1.22}$	$72.20_{\pm0.78}$	66.01
3000 Training Steps							
FVT	$\mathbf{58.44}_{\pm0.49}$	$\mathbf{51.47}_{\pm0.42}$	$62.70_{\pm1.00}$	$69.53_{\pm1.14}$	$\mathbf{83.29}_{\pm1.20}$	$69.35_{\pm0.80}$	65.79
SAVA	$57.82_{\pm0.49}$	$51.08_{\pm0.42}$	$\mathbf{63.17}_{\pm1.00}$	$\mathbf{69.78}_{\pm1.14}$	$81.73_{\pm1.24}$	$\mathbf{74.15}_{\pm0.76}$	**66.29**
LAPT	$59.35_{\pm0.49}$	$52.94_{\pm0.42}$	$62.96_{\pm1.00}$	$69.72_{\pm1.14}$	$82.98_{\pm1.21}$	$71.77_{\pm0.78}$	66.62

Table 4: 0-shot results over **Italian** translated benchmarks for *Llama-3.1-8B* adapted models.

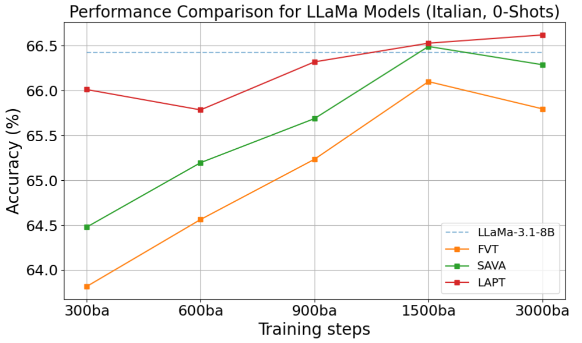

Figure 3: Average performance of *Llama-3.1-8B* based models during training on **Italian** translated benchmarks. The average was calculated over six datasets.

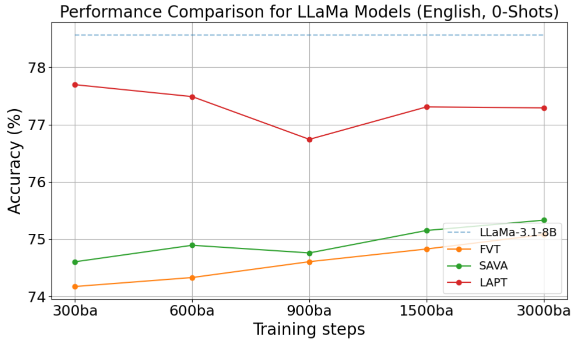

Figure 5: Average performance of *Llama-3.1-8B* based models during training on **English** benchmarks. The average was calculated over six datasets.

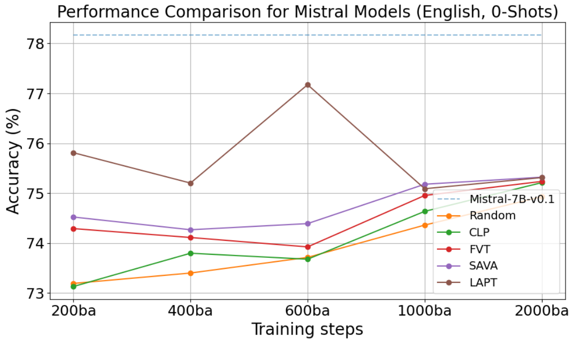

Figure 4: Average performance of *Mistral-7B-v0.1* based models during training on **English** benchmarks. The average was calculated over six datasets.

erage score at the end of the adaptation process. All the adapted models diminish in performance compared to the base one in the English language.

Figure 5 reports the average scores of *Llama-3.1-8B* models on English benchmarks during training. In this setting, *LAPT* maintains higher performance on average; intuitively, this could be attributed to the larger vocabulary of *Llama-3.1-8B* (75% bigger), which enables better performance during language adaptation, avoiding catastrophic forgetting of the source language.

For both models the *SAVA* approach leads the model to achieve slightly higher performance in the source language. Appendix E reports more detailed results of evaluation over English benchmarks.

5.2 Generative Setting

Multi-choice benchmarking based on perplexity scoring has its own limitations (Wang et al., 2024). To further test our models, we evaluate them on two generative tasks: Machine Translation (MT), IT-EN and EN-IT, and Italian Question Answering.

We report COMET-22 (Rei et al., 2022) for the MT benchmark and RougeL (Lin, 2004) for the Question Answering task.

Looking at the MT results, in Table 5, we observe that the adapted *Mistral-7B-v0.1* models achieve excellent performance, outperforming those of the base model. The vocabulary adapted models reach very good results in the English-to-Italian direction, where generation of Italian text is involved. Our findings indicate that *SAVA* and *FVT* emerge as the most effective vocabulary adaptation heuristics in this context. As shown in Table 6, a similar trend is observed with *Llama-3.1-8B*, where adapted models perform competitively with the

	FLoRes		SQuAD-it
Model	**EN-IT**	**IT-EN**	**RL**
Mistral-7B-v0.1	86.57	87.75	68.92
200 Training Steps			
Random	86.67	87.37	62.1
FVT	87.08	87.55	65.47
CLP	86.58	87.31	64.25
SAVA	**87.30**	**87.59**	**65.66**
LAPT	87.41	87.92	67.35
2000 Training Steps			
Random	88.01	**87.92**	64.83
FVT	88.29	87.90	66.18
CLP	88.21	87.79	65.99
SAVA	**88.31**	87.87	**67.20**
LAPT	88.13	88.02	66.92

Table 5: 5-shot results for *Mistral-7B-v0.1* of FLoRes where COMET-22 is reported and 2-shot results for SQuAD-it where RougeL is reported.

	FLoRes		SQuAD-it
Model	**EN-IT**	**IT-EN**	**RL**
Llama-3.1-8B	87.59	88.08	69.21
300 Training Steps			
FVT	87.32	**87.65**	68.54
SAVA	**87.39**	87.58	**68.70**
LAPT	87.82	87.95	67.91
3000 Training Steps			
FVT	88.05	88.02	68.84
SAVA	**88.12**	**88.04**	**69.05**
LAPT	88.11	88.05	66.69

Table 6: 5-shot results for *Llama-3.1-8B* of FLoRes where COMET-22 is reported and 2-shot results for SQuAD-it where RougeL is reported.

base model, while *SAVA* and *FVT* reach the same performance as those of *LAPT*.

Regarding the results in the SQuAD-it task, Tables 5 and 6 show that *SAVA* attains very good performance, beating other heuristics and the *LAPT* approach for both model types, reaching inline performance equal to that of the base model for *Llama-3.1-8B*.

5.3 Training Loss

Important observations can be made concerning the loss trajectories. Figure 6 reports the *Mistral-7B-v0.1* plots, and we can notice significant differences between the various heuristics in the early stages of the training. The *SAVA*-model emerges as the better-adapted one, right from the start, particularly

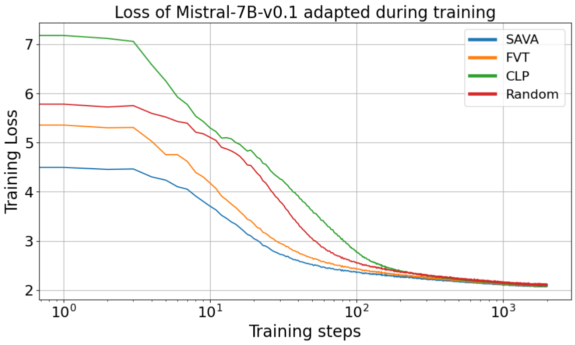

Figure 6: Loss during continual training of *Mistral-7B-v0.1* models.

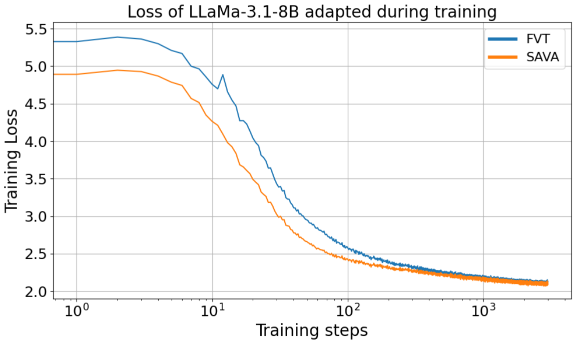

Figure 7: Loss during continual training of *Llama-3.1-8B* models.

when compared to the *CLP* and *Random* models. Notably, *CLP* appears to lag behind *Random* initially. Looking at *Llama-3.1-8B* losses, in Figure 7 we can see that the two heuristics exhibit similar trajectories, although *SAVA* still achieves a lower loss from the outset.

6 Differences in the Embedding Structure

To better understand the impact of different vocabulary adaptation techniques, we analyze similarities in intra-model and inter-model embedding spaces. Specifically, we examine how different adaptations influence the structural alignment of embeddings in comparison to a reference model (intra-model similarity) and how the embedding spaces of different adapted models compare to each other (inter-model similarity).

To measure the similarity between two embedding spaces, we rely on the technique introduced by Moschella et al. (2023). Specifically, we randomly select 128 non-prefix tokens and 128 prefix tokens from V_t to compute relative embedding representations, resulting in a total of 256 anchor tokens.[7]

[7] Non-prefix tokens refer to complete words or sub-words

	Mistral-7B-v0.1		*Llama-3.1-8B*	
Model	**@0ba**	**@2000ba**	**@0ba**	**@3000ba**
Random	29.68	31.67	-	-
FVT	33.65	35.30	33.23	33.49
CLP	41.10	42.84	-	-
SAVA	**44.81**	**45.33**	**41.84**	**42.02**

Table 7: Similarity scores between *Mistral-7B-v0.1* adapted models and *Minerva-3B* (left) and between *Llama-3.1-8B* and *Minerva-3B* (right) at the beginning and at the end of the training.

For each model, we then adjust the representation of each token relative to these anchors, calculating each dimension as the projection onto the selected anchors. Subsequently, we compute the cosine similarity based on this relative representation across the models and average the results to obtain an overall similarity score between the two distinct models.

Intra-model similarity Intuitively, a well-adapted model should align with *Minerva-3B*, as it serves as a strong reference for the target language. Similarly to our setting, *Minerva-3B* is pretrained on balanced Italian-English data from CulturaX. In Table 7, we present the similarity scores between adapted models and *Minerva-3B*. Notably, *CLP* and *SAVA* achieve higher similarity scores than other approaches. This outcome is to be expected, as both *CLP* and *SAVA* leverage *Minerva-3B*'s embedding space. Interestingly, *SAVA* not only attains a structure that is more similar to that of *Minerva-3B* (+3.7), but also demonstrates superior performance, as was also the case in previous sections.

Inter-model similarity To gain deeper insights into the differences in learned embedding structures, Figure 8 presents the similarity scores between *Mistral-7B-v0.1* variants adapted using the specified techniques. We compare the models at the end of the continual training. The analysis shows a high similarity between models, but differences of up to 10% in the relative representations reveal structural variations in the encoded information. This analysis suggests that, even after intensive training, the adapted models do not converge to the same representation.

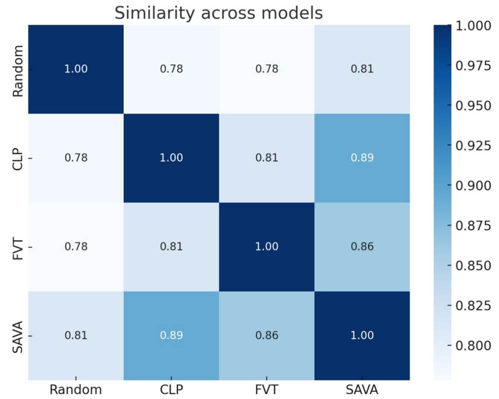

Figure 8: Similarity across models after continual training on 12B tokens.

7 Conclusions

In this work, we extensively explored various techniques to adapt English-focused LLMs, i.e. *Mistral-7B-v0.1* and *Llama-3.1-8B*, to the Italian language. We introduced a novel heuristic called *SAVA* which leverages the embedding structure of a smaller, native Italian Language Model, *Minerva-3B*. We discovered that adapting the vocabulary of English LLMs leads to significant improvements in language encoding, reducing the number of generated tokens by 25% for *Mistral-7B-v0.1* and 16% for *Llama-3.1-8B*. Regarding *Llama-3.1-8B* we pruned nearly 1 billion parameters by optimizing its vocabulary, removing approximately 75% of the original tokens. Our evaluation revealed performance differences across the vocabulary adaptation heuristics, by means of a thorough analysis during the continual training phase. We show that linguistic capabilities can be restored with relatively few training steps—*Mistral-7B-v0.1* reached base model performance after processing 2 billion tokens. Additionally, the *SAVA* heuristic demonstrated strong performance on downstream tasks, with *SAVA*-adapted models reaching faster convergence during continual training. Furthermore, the embedding structure of *SAVA* exhibited closer alignment with the helper model compared to other analyzed heuristics.

This work opens several research directions. One key area of interest will be to evaluate how the *SAVA* approach scales across languages, particularly in mid- and low-resource settings. Understanding how different heuristics perform with a small number of continual training steps in such scenarios is crucial. Additionally, since *Minerva-7B* was not available at the time of writing, a logical next step would be to utilize it as a helper model.

that do not start a sequence (e.g., “cat” in “concatenate”), while prefix tokens initiate a word or sub-word.

8 Limitations

We investigated the adaptation of English-first LLMs to the Italian language with a focus on adapting the vocabulary and the tokenizer to match the performance of continually trained models while achieving lower fertility and thus higher efficiency in the target language.

We limited our training data to the CulturaX dataset, which consists of cleaned web-crawled data. Incorporating higher-quality datasets could improve the models' performance in the target language.

We limited our analysis to two distinct decoder-only Large Language Models: *Mistral-7B-v0.1* and *Llama-3.1-8B*. For a more comprehensive study, additional English-first models could be tested. However, the aforementioned two models are among the best performing ones in their parameter count. Furthermore, we chose to focus on just two models due to the extensive continual training we had to perform, as such training requires considerable computational resources.

We evaluated the adapted models on automatically translated datasets for multiple-choice tasks and open-ended question answering. Specifically, Hellaswag, MMLU, Arc Easy, PIQA, SCIQ, and BOOLQ were translated using *Tower-Instruct-v0.2*, an open-source solution for automatic translation that, at the time of writing, represents the state of the art in open Machine Translation models. For generative tasks, SQuAD-it was translated using a semi-automatic approach.

We acknowledge that relying on automatically translated benchmarks may have introduced some noise, potentially obscuring certain abilities or issues in the models' comprehension of Italian texts. This limitation was beyond our capabilities to resolve since no well-structured Italian native benchmarks exist. Another limitation was using only two generative benchmarks, where we observed slightly different results for the adapted models. In the generative setting, SAVA generally outperformed other methods, while LAPT models did not consistently deliver the best average performance on downstream tasks.

Future work should aim to explore the capabilities of vocabulary-adapted models in generative tasks and investigate how a model's fertility over target language influences downstream performance.

9 Ethics Statement

We primarily conduct experiments in the Italian language. This approach is aimed at addressing the practical challenges of working with Italian, a language that is underrepresented in the NLP field. Our continual training is performed on data collected from open web sources, specifically through the CulturaX dataset. Since large-scale datasets used for pretraining can include personal and sensitive information, it is crucial to carefully assess such content before deploying models in real-world applications. Another key consideration is the use of existing monolingual or multilingual models as starting points, rather than training new models from scratch. This can introduce biases from the original pretraining data, potentially causing the model to reflect behaviors and cultural influences from other languages rather than those of the target language community.

Acknowledgments

Edoardo Barba and Alessio Miaschi are fully funded by the PNRR MUR project PE0000013-FAIR. Roberto Navigli and Felice Dell'Orletta acknowledge the support of the PNRR MUR project PE0000013-FAIR. Partially financed by the European Union - NextGenerationEU through the Italian Ministry of University and Research under PNRR - PRIN 2022 (2022EPTPJ9) "WEMB: Word Embeddings from Cognitive Linguistics to Language Engineering and back" and by the PNRR project ITSERR (CUP B53C22001770006). We acknowledge the support of the ISCRA project TRAVEL (HP10CY9V7K) for awarding access to the LEONARDO supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy) and thank Giuseppe Fiameni for his support.

References

Pierpaolo Basile, Elio Musacchio, Marco Polignano, Lucia Siciliani, Giuseppe Fiameni, and Giovanni Semeraro. 2023. Llamantino: Llama 2 models for effective text generation in italian language. *Preprint*, arXiv:2312.09993.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1324–1334, Online. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Y. Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, arXiv:2207.04672.

Danilo Croce, Alexandra Zelenanska, and Roberto Basili. 2018. Neural learning for question answering in italian. In *AI*IA 2018 – Advances in Artificial Intelligence*, pages 389–402, Cham. Springer International Publishing.

Zoltan Csaki, Bo Li, Jonathan Lingjie Li, Qiantong Xu, Pian Pawakapan, Leon Zhang, Yun Du, Hengyu Zhao, Changran Hu, and Urmish Thakker. 2024. SambaLingo: Teaching large language models new languages. In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 1–21, Miami, Florida, USA. Association for Computational Linguistics.

Wietse de Vries and Malvina Nissim. 2021. As good as new. how to successfully recycle English GPT-2 to make models for other languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 836–846, Online. Association for Computational Linguistics.

Konstantin Dobler and Gerard de Melo. 2023. FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13440–13454, Singapore. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Leonidas Gee, Andrea Zugarini, Leonardo Rigutini, and Paolo Torroni. 2022. Fast vocabulary transfer for language model compression. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 409–416, Abu Dhabi, UAE. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. 2022. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. IndoBERTweet: A pretrained language model for Indonesian Twitter with effective domain-specific vocabulary initialization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10660–10668, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yihong Liu, Peiqin Lin, Mingyang Wang, and Hinrich Schuetze. 2024. OFA: A framework of initializing unseen subword embeddings for efficient large-scale multilingual continued pretraining. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1067–1097, Mexico City, Mexico. Association for Computational Linguistics.

Valentino Maiorca, Luca Moschella, Antonio Norelli, Marco Fumero, Francesco Locatello, and Emanuele Rodolà. 2024. Latent space translation via semantic alignment. *Advances in Neural Information Processing Systems*, 36.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Cristian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, and et al. 2024. Gemma: Open models based on gemini research and technology. volume abs/2403.08295.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Luca Moroni, Simone Conia, Federico Martelli, and Roberto Navigli. 2024. ITA-Bench: Towards a more comprehensive evaluation for Italian LLMs. In *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. 2023. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2024. CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4226–4237, Torino, Italia. ELRA and ICCL.

Riccardo Orlando, Luca Moroni, Pere-Lluís Huguet Cabot, Edoardo Barba, Simone Conia, Sergio Orlandini, Giuseppe Fiameni, Roberto Navigli, et al. 2024. Minerva llms: The first family of large language models trained from scratch on italian data. In *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*.

Malte Ostendorff and Georg Rehm. 2023. Efficient language model training through cross-lingual and progressive transfer learning. *arXiv preprint arXiv:2301.09626*.

Marco Polignano, Pierpaolo Basile, and Giovanni Semeraro. 2024. Advanced natural-based interaction for the italian language: Llamantino-3-anita. *Preprint*, arXiv:2405.07101.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024. "my answer is C": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7407–7416, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. *CoRR*, abs/2411.12372.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106, Copenhagen, Denmark. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A SAVA Training of the mapping function

To implement the SAVA methods, we first need to train the linear mapping function, ϕ. For this, we use the `SGDAffineAligner` method provided in the **latentis** library[8].

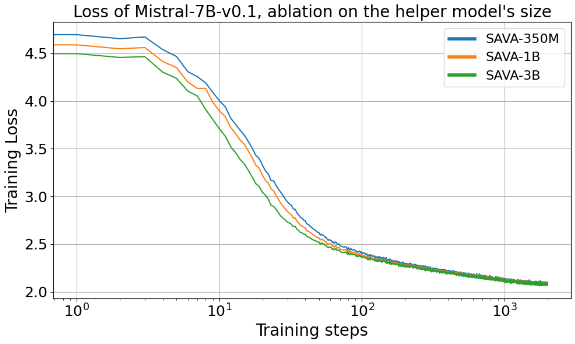

Figure 9: Loss during continual training of Mistral models.

After collecting the token representation pairs from the intersection, we train the linear mapping using the `ADAM` optimizer with `MSE Loss`, setting the learning rate to 10^{-3} and running the optimization for 1000 steps.

To enhance training stability, we first apply *standard scaling* and *L2 normalization* to the token representations before learning ϕ. After training, we apply the inverse scaling to restore the original distribution before incorporating the results into the adapted model.

B Ablation experiments on the SAVA method

In this section we analyze some ablation studies over the *SAVA* method. We analyzed the impact on the helper model's size, using the two smaller models of Minerva's family, *Minerva-350M* and *Minerva-1B*, which have, respectively, 350M and 1B parameters. In Figure 9 the training loss of *Mistral-7B-v0.1* adapted using *SAVA* with different helper models is reported. From the plot we can see that the dimension of the helper model does not have a huge impact on the loss trajectory. An orthogonal experiment was conducted to ablate the number of tokens used to learn the mapping ϕ, in Figure 10 the loss for *Mistral-7B-v0.1* adapted with *SAVA* relying on different number of tokens in $V_t \cap V_s$, over *Minerva-3B* is reported. We observe that using more tokens leads to a faster convergence of the training loss. From the plots we can see that reducing the number of tokens has a greater impact than reducing the model size, especially for the setting with two thousand tokens.

[8] `https://github.com/Flegyas/latentis`

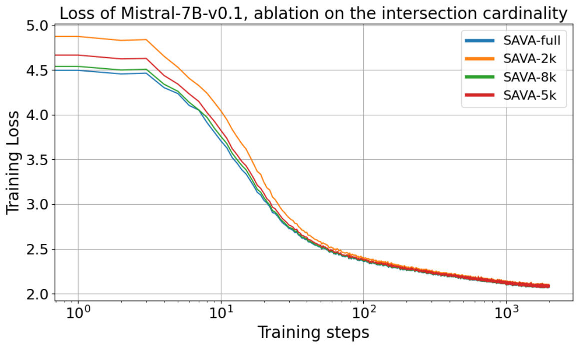

Figure 10: Loss during continual training of Mistral models.

C Training Resources and Environmental Impact

Experiments were conducted using the LEONARDO Italian Supercomputer, which has a carbon efficiency of 0.432 kgCO$_2$eq/kWh. A cumulative of 50000 hours of computation was performed on hardware of type A100 SXM4 80 GB (TDP of 400W).

Total emissions are estimated to have been 8640 kgCO$_2$eq of which 0 percent were directly offset. These emissions were split roughly into 95% for continual training and 5% for evaluation.

This is an approximate estimate since the computation was done on LEONARDO custom hardware which is not available in the tool used for the estimation .

D Generation setting

We tested our adapted models on two downstream tasks in a generative setting, machine translation and question answering. We tested the models in few-shot setting relying on the in-context capabilities of evaluated models, without any fine-tuning step to the specific task. We relied on the vLLM library (Kwon et al., 2023) to afford prompting generation, specifically we changed the default parameters with `temperature=0` and `max_tokens=512`.

After a comprehensive number of trials we noticed that the prompting strategy had a huge impact, while the order between the models remained unchanged. We report the prompts used for FLoRes and SQuAD-it tasks in Tables 8 and 9, respectively.

E English Results on Multi-choice benchmarks

In this section, we present a detailed analysis of the evaluation results on English benchmarks. Table 10 reports the performance of *Mistral-7B-v0.1* on six multiple-choice benchmarks. From this table, we observe that *SAVA* and *FVT* achieve higher task-wise scores early in the adaptation process. A similar trend is evident for the *Llama-3.1-8B* adapted models, as shown in Table 11, where the *SAVA* technique yields higher average scores than *FVT*, at the beginning and at the end of training. For both models, per-task scores remain below the base model's performance. However, incorporating a portion of English data during adaptation prevents catastrophic forgetting when transitioning towards the Italian language.

Prompt EN-IT	**Prompt IT-EN**
Traduci dall'Inglese all'Italiano Text: I love you so much. Translation: Ti amo così tanto.	Translate from Italian to English Text: Ti amo così tanto. Translation: I love you so much.

Table 8: Prompts used for machine translation task

Italian Prompt
Contesto: Il terremoto del Sichuan del 2008 o il terremoto del Gran Sichuan, misurato a 8.0 Ms e 7.9 Mw, e si è verificato alle 02:28:01 PM China ... Domanda: In quale anno si è verificato il terremoto nel Sichuan? Risposta: 2008

Table 9: Used prompts for question answering task

Model	**Hellaswag**	**MMLU**	**Arc Easy**	**PIQA**	**SCIQ**	**BOOLQ**	**AVG**
Mistral-7B-v0.1	$75.98_{\pm 0.44}$	$57.19_{\pm 0.42}$	$78.55_{\pm 0.94}$	$83.84_{\pm 0.94}$	$95.82_{\pm 0.80}$	$77.64_{\pm 0.78}$	78.17
			200 Training Steps				
Random	$72.29_{\pm 0.44}$	$51.59_{\pm 0.42}$	$69.55_{\pm 0.95}$	$81.73_{\pm 0.96}$	$89.97_{\pm 0.97}$	$74.03_{\pm 0.76}$	73.19
FVT	$72.35_{\pm 0.44}$	$53.04_{\pm 0.42}$	$73.08_{\pm 0.92}$	$82.60_{\pm 0.94}$	$92.48_{\pm 0.85}$	$72.20_{\pm 0.78}$	$\underline{74.29}$
CLP	$72.59_{\pm 0.44}$	$52.02_{\pm 0.42}$	$70.16_{\pm 0.94}$	$81.55_{\pm 0.96}$	$89.66_{\pm 0.98}$	$72.81_{\pm 0.77}$	73.13
SAVA	$72.81_{\pm 0.44}$	$53.21_{\pm 0.42}$	$74.28_{\pm 0.94}$	$82.47_{\pm 0.96}$	$92.79_{\pm 0.83}$	$71.59_{\pm 0.78}$	**74.52**
LAPT	$74.13_{\pm 0.43}$	$55.05_{\pm 0.42}$	$75.23_{\pm 0.89}$	$84.02_{\pm 0.91}$	$94.46_{\pm 0.73}$	$71.98_{\pm 0.78}$	75.81
			2000 Training Steps				
Random	$72.18_{\pm 0.44}$	$52.11_{\pm 0.42}$	$73.6_{\pm 0.91}$	$82.72_{\pm 0.94}$	$93.21_{\pm 0.81}$	$75.77_{\pm 0.74}$	74.93
FVT	$73.28_{\pm 0.44}$	$52.96_{\pm 0.42}$	$74.76_{\pm 0.90}$	$81.91_{\pm 0.95}$	$94.05_{\pm 0.76}$	$74.46_{\pm 0.76}$	$\underline{75.23}$
CLP	$73.37_{\pm 0.44}$	$52.48_{\pm 0.42}$	$74.07_{\pm 0.90}$	$82.47_{\pm 0.94}$	$94.05_{\pm 0.76}$	$74.83_{\pm 0.75}$	75.21
SAVA	$73.02_{\pm 0.44}$	$52.91_{\pm 0.42}$	$74.67_{\pm 0.90}$	$82.29_{\pm 0.94}$	$94.46_{\pm 0.73}$	$74.58_{\pm 0.76}$	**75.32**
LAPT	$74.26_{\pm 0.43}$	$51.18_{\pm 0.42}$	$73.9_{\pm 0.91}$	$83.65_{\pm 0.92}$	$94.67_{\pm 0.72}$	$74.22_{\pm 0.76}$	75.31

Table 10: 0-shot results over English benchmarks for *Mistral-7B-v0.1* adapted models.

Model	**Hellaswag**	**MMLU**	**Arc Easy**	**PIQA**	**SCIQ**	**BOOLQ**	**AVG**
LLaMa-3.1-8B	$74.21_{\pm 0.43}$	$62.19_{\pm 0.42}$	$77.60_{\pm 0.47}$	$83.03_{\pm 0.93}$	$93.94_{\pm 0.77}$	$80.42_{\pm 0.69}$	78.56
			300 Training Steps				
FVT	$72.35_{\pm 0.44}$	$58.22_{\pm 0.42}$	$69.55_{\pm 0.95}$	$81.30_{\pm 0.97}$	$92.27_{\pm 0.86}$	$71.34_{\pm 0.79}$	74.17
SAVA	$72.72_{\pm 0.44}$	$58.19_{\pm 0.42}$	$70.75_{\pm 0.94}$	$81.79_{\pm 0.96}$	$92.90_{\pm 0.83}$	$71.28_{\pm 0.79}$	**74.60**
LAPT	$74.35_{\pm 0.43}$	$61.74_{\pm 0.41}$	$76.14_{\pm 0.88}$	$83.21_{\pm 0.93}$	$94.05_{\pm 0.76}$	$76.69_{\pm 0.73}$	77.69
			3000 Training Steps				
FVT	$73.02_{\pm 0.44}$	$57.85_{\pm 0.42}$	$72.13_{\pm 0.93}$	$82.04_{\pm 1.15}$	$92.90_{\pm 0.83}$	$72.53_{\pm 0.78}$	75.07
SAVA	$72.86_{\pm 0.44}$	$57.94_{\pm 0.42}$	$72.78_{\pm 0.92}$	$81.79_{\pm 0.96}$	$93.31_{\pm 0.80}$	$73.30_{\pm 0.77}$	**75.33**
LAPT	$74.40_{\pm 0.43}$	$60.50_{\pm 0.42}$	$75.32_{\pm 0.89}$	$82.47_{\pm 0.94}$	$93.63_{\pm 0.78}$	$77.43_{\pm 0.73}$	77.29

Table 11: 0-shot results over English benchmarks for *Llama-3.1-8B* adapted models.