Improving Language Models Trained on Translated Data with Continual Pre-Training and Dictionary Learning Analysis

Sabri Boughorbel, Md Rizwan Parvez, Majd Hawasly
Qatar Computing Research Institute
Hamad Bin Khalifa University, Doha, Qatar
{sboughorbel, mparvez, mhawasly}@hbku.edu.qa

Abstract

Training LLMs in low resources languages usually utilizes machine translation (MT) data augmentation from English language. However, translation brings a number of challenges: there are large costs attached to translating and curating huge amounts of content with high-end machine translation solutions; the translated content carries over cultural biases; and if the translation is not faithful and accurate, the quality of the data degrades causing issues in the trained model. In this work, we investigate the role of translation and synthetic data in training language models. We translate TinyStories, a dataset of 2.2M short stories for 3-4 year old children, from English to Arabic using the open NLLB-3B MT model. We train a number of story generation models of size 1M-33M parameters using this data. We identify a number of quality and task-specific issues in the resulting models. To rectify these issues, we further pre-train the models with a small dataset of synthesized high-quality stories generated by a capable LLM in Arabic, representing 1% of the original training data. We show, using GPT-4 as a judge and dictionary learning analysis from mechanistic interpretability, that the suggested approach is a practical means to resolve some of the translation pitfalls. We illustrate the improvement through case studies of linguistic and cultural bias issues.

1 Introduction

Large Language Models (LLMs) have transformed the landscape of natural language processing (NLP) significantly. With the realization of unprecedented capabilities, the availability of underlying large training corpora and well-established language model training pipelines has shifted the focus in NLP research from defining linguistic inductive biases to the collection and curation of extensive text datasets (Soldaini et al., 2024; Penedo et al., 2024; Computer, 2023; Mehta et al., 2024). The recent trend showed an increase in focus on data curation and augmentation compared to innovation in model architecture or training paradigms (Brown et al., 2020; Touvron et al.,

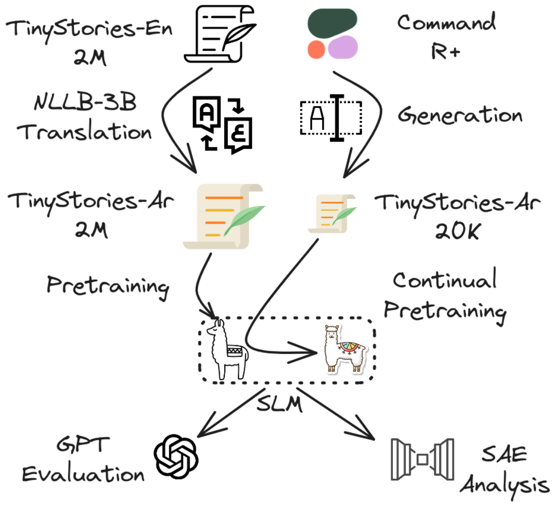


Figure 1: The proposed TinyStories Arabic dataset is formed by translating 2M tiny stories from English to Arabic using NLLB-3B and synthesizing 20K Arabic tiny stories using Command R+ LLM. The former data is used to pre-train small language models (SLMs) with different architectures. The latter is used for continual pre-training. The models are qualitatively and quantitatively evaluated using a GPT judge. Further we train Sparse Auto-Encoder (SAE) on a selected SLM to analyze the model behavior.

Proceedings of The Second Arabic Natural Language Processing Conference, pages 73–88
August 16, 2024 ©2024 Association for Computational Linguistics

2023). However, emergent capabilities in LLMs are noticeable for models of very large sizes such as 10B parameters or above and most of smaller LLMs of 200M-3B parameters have shown limited abilities in reasoning, fact recall and coherent long generation (Schaeffer et al., 2024). While these models are typically trained with the same data used to train their larger and more capable siblings, the drops in capabilities of these small LLM are usually attributed to their reduced learnability or scale.

However, newer small models have been recently shown to compete with 10x bigger models in challenging tasks. Examples include MiniMA (Zhang et al., 2023) and the Phi family (Abdin et al., 2024; Li et al., 2023). These small language models (SLMs) offer faster training and lower deployment cost at the expense of being more task-specific and less general-use, which is not an issue for application-oriented language models. The key ingredient for the success of these smaller models seems to be the use of sophisticated and aggressive data curation, and high-quality synthetic data generated by bigger LLMs.

This new trend, however, does not map equally to languages that are not as privileged with huge amounts of high quality content (or content in quantities that allow aggressive filtering), nor the availability of strong models that can be employed to generate diverse synthetic data in substantial quantities in a cost-effective manner, as is the situation with Arabic and many other low-resourced languages (Thompson et al., 2024).

A commonly-adopted workaround to the issue of data shortage is to turn to machine translation (MT) to benefit from the available content in English, which is evident in the data mixtures of the more capable Arabic models, e.g. Jais (Sengupta et al., 2023). However, the use of machine translation does not come without pitfalls. Specifically, 1) cultural biases that are stored within a language corpus get imported when translation is used, leading to misaligned models, e.g. (Holmström et al., 2023) and 2) based on the quality of the translation, certain linguistic intricacies of languages might not be respected (Zhang and Toral, 2019), leading to degradation in the quality of the data and thus degradation in the capability of the final model with regard to the quality of the target language.

In this work, we study this phenomenon using the recently-released TinyStories (Eldan and Li, 2023), a synthetic dataset introduced to explore emergent properties in small language models. TinyStories comprises 2.2M short stories in English of about 200 words generated by GPT-3.5 and GPT-4. The relatively small models trained on TinyStories have shown interesting capabilities in generating coherent and creative short stories with correct grammar. We translate TinyStories to Arabic using the open-source translation model NLLB-3B to simulate medium-quality translation. We train models with different sizes using the Arabic-translated TinyStories, and benchmark the trained models against several other Arabic LLMs using GPT-4 as a judge following previous works (Zheng et al., 2024) in the task of story generation using three metrics: grammar correctness, consistency with the provided context, and creativity. We also identify some cultural and linguistic issues that arise from the translation. Then, we synthesize a high quality small dataset of 20K stories to explore the efficacy of continual pre-training in recovering from the issues brought along by low-quality translated data.

The contributions of this work are as follow:

- We investigate the degradation of language models when using translated data of medium quality in training. We identify linguistic issues and cultural biases, which we address by further pre-training the models with a limited amount of high-quality synthetic data.
- We create a dataset with an Arabic version of TinyStories through medium quality translation, plus a small high-quality synthetic data which we will release as open-source to facilitate studying translation issues.
- We provide a comparative analysis before and after refinement using dictionary learning methods from mechanistic interpretability to assess the effects of continual pre-training with high-quality data.

The rest of the paper is organized as follows: First, we describe how we prepare training data. Then we discuss the limitations of translated data. In Section 3.1, we show how continual pre-training with small amount of high-quality improve the models.

2 Story Generation SLMs

2.1 TinyStories dataset

First we give a brief description of the original TinyStories dataset (Eldan and Li, 2023). It consists of 2.2M short stories for 3-4 year old kids generated by GPT-3.5 and GPT-4. A set of 1500 basic words consisting of nouns, verbs and adjectives are initially collected that can be easily understood by a 3-year-old child. For each story, a verb, a noun and an adjective are randomly selected from the list. GPT models are prompted to generate a story including all the selected words, and that respects certain storytelling features, like a certain plot or a way of ending as could be seen in Table 6 in the Appendix.

2.2 Translated TinyStories

We translated the dataset using an open source machine translation model that belongs to the 'No Language Left Behind' project (Costa-jussà et al., 2022) which is intended primarily for 200 low-resource languages, offering a family of five models with sizes ranging from 600M parameters to 54B parameters. NLLB has good Arabic translation performance compared to other solutions (Tiedemann et al., 2023; Kudugunta et al., 2024). We chose NLLB-3B, a model that can be deployed locally with reasonable hardware requirements, e.g, a single GPU with 16GB memory. The 3B size was chosen with practicality in mind, and to simulate the use of medium quality translation when preparing data on a large scale for language model training.

2.3 Evaluation Method

We followed a similar evaluation approach to the previous work on TinyStories (Eldan and Li, 2023). We translate the test set of 44 manually-picked story completion prompts from English to Arabic using GPT-4. An example story completion prompt from the test set could be seen in Appendix A.2. The choice of GPT-4 as a translator was to obtain a high-quality test set, and to ensure that the test data is out-of-distribution to the training data from the translation perspective.

During evaluation, GPT-4 is also used as a judge to assess the story completion of the tested models. The LLM judge is asked to pay extra attention to the initial sentence that is cut short, and to assess three main qualities: correctness of grammar, creativity of the story, and consistency to the prompt and the story details. Each of the metrics is given an integer score between 0 and 10. For each model, two completions with temperature 1 are generated for each test prompt, then average scores are reported.

2.4 Model Training

We use the translated dataset to train a number of small language models. We consider the same model size and configuration as in the original TinyStories family[1]. The only difference was in replacing GPT-NeoX architecture with Llama-2. This choice was based on the success and wider adoption of Llama to train LLMs. The main contrast between the two architectures is that LLama-2 uses gated MLP layers, grouped-query attention and an RMS normalization, whereas GPT-NeoX uses parallel attention MLP and Layer-Norm normalization (Zhang and Sennrich, 2019; Liu et al., 2021; Ainslie et al., 2023; Black et al., 2022). In the chosen model family, the number of heads is fixed, whereas the number of layers and residual stream (latent space) are varied. In 1M, 3M and 8M, the number of layers is kept constant at 8, while the dimension of residual stream is increased. In the remaining models, the increase in the number of layers is compensated by a reduction in the residual stream dimension. The goal is to maintain the model size in the range of 20M-30M parameters. Table 1 provides a summary of the architecture of trained models.

For SLM training, the choice of tokenizer is of utmost importance. Since the embedding layer

[1] https://huggingface.co/roneneldan

	Layers	Heads	Hidden size
33M-ar	4	16	768
28M-ar	8	16	512
2L-33M-ar	2	16	1024
1L-21M-ar	1	16	1024
8M-ar	8	16	256
3M-ar	8	16	128
1M-ar	8	16	64

Table 1: Architecture of trained models

can consume a large proportion of trainable parameters, it is crucial to carefully choose the vocabulary size and tokens. Existing tokenizers suited for Arabic language models usually include English or are over-sized for our purpose of training SLMs. We decide to train a byte-pair encoding (BPE) tokenizer on the translated training data. We experimented with different vocabulary sizes (8k, 16k, 32k and 85k). The last two choices are motivated by the Llama-2 and Jais tokenizers. We trained a reference model (33M) with the different tokenizers to evaluate the role of vocabulary size on performance. As shown in Table 2, we selected a vocabulary size of 32k given it performed the highest.

Vocab size	Grammar	Creativity	Consistency
8k	2.97	3.97	2.69
16k	4.64	4.72	4.21
32k	**6.83**	**7.01**	**7.20**
85k	3.02	3.54	1.77

Table 2: Evaluation of different vocabulary sizes on the performance of a selected SLM.

For model training, we used the English data and published benchmarks as a means to calibrate the training process, see Appendix B.1. We use AdamW optimizer with constant learning rate of $5 \, . \, 10^{-5}$ as in TinyStories paper with 5% steps for warm-up (Eldan and Li, 2023).

The training and validation loss of the training can be seen in Appendix B.2. The shape of the loss curves indicate that the training settings are adequate. Models with lower loss have better benchmarking performance as we will discuss later. The convergence loss for models trained on Arabic data are higher than the models trained on English data. One justification could be that the noise introduced by translation makes the task of next token prediction more difficult than the original task.

Model	Grammar	Creativity	Consistency
28M-ar	6.80	**7.32**	**7.28**
33M-ar	**6.83**	7.01	7.20
2L-33M-ar	6.52	7.01	7.00
1L-21M-ar	5.87	6.11	5.96
8M-ar	6.47	6.84	6.70
3M-ar	5.82	6.48	5.79
1M-ar	4.37	5.11	3.43

Table 3: Benchmarking the trained models using GPT-4 as a judge on grammar, creativity and consistency

Table 3 shows the GPT-4 scores on the three benchmarking dimensions (grammar, creativity and consistency) for the trained models on translated TinyStories. The best performance is achieved between the 28M and 33M models, and we will use the latter for the rest of this paper. An example completion of the 33M-ar model could be found in Appendix A.2.

We also compare the results of our trained SLMs with some state-of-the-art Arabic or multilingual LLMs of sizes 200M-13B (Radford et al., 2019; Antoun et al., 2021; Koubaa et al., 2024; Workshop et al., 2023; Kamal Eddine et al., 2022; Elmadany et al., 2023; Sengupta et al., 2023) in Table 4.

3 Navigating Pitfalls of Translated Data

There are issues faced when we train models with medium-quality translation: 1) **Cultural biases**: translated data carries over the culture of the source data into the target language, which might not be desirable. We choose in this work one example from this category to focus on, which is that the translated stories come with English person names, leading to an Arabic model that can only generate stories with such names; 2)

Model	Grammar	Creativity	Consistency
gpt2-small	3.23	2.43	1.38
aragpt2	3.02	2.84	1.76
arabian-gpt	3.86	4.27	2.65
bloom-1b1	5.34	4.75	4.00
AraBart	7.07	5.03	5.96
33M-ar	6.83	**7.01**	7.20
AraT5v2	8.03	6.69	8.33
jais-13b	**8.10**	6.89	**8.43**

Table 4: Comparing the 33M-ar model to some state-of-the-art Arabic models using GPT-4 as a judge

Models	Grammar	Creativity	Consistency
28M-ar	6.80	7.32	7.28
33M-ar	6.83	7.01	7.20
33M-ar-CP	7.08	7.21	7.26

Table 5: Performance comparison of the continued pre-trained model (33M-ar-CP) and the best two base models trained only on translation data (28M-ar and 33M-ar). We choose the largest model as baseline in this experiment.

Grammatical and style issues: languages differ in how they express the same sentence, and weaker translations might fail to correct for nuanced style issues. We choose a single example related to how the speaker could follow the saying in English but not in Arabic. For example, the direct Arabic translation of the English sentence "`'I'm happy!', said Tim.`" to

„أنا سعيد!„، قال تيم.

is not appropriate.

3.1 Model refinement with continual pre-training

In order to tackle some of the MT issues, we test whether the trained models can be improved with a second stage of training utilizing a small amount of high-quality data that does not have the aforementioned issues. This practical and cost-effective approach is somewhat similar to curriculum learning where LLMs are exposed to different levels of complexity at different stages of training (Zhou et al., 2020; Chang et al., 2021).

3.1.1 Synthetic Generation

We synthesized an additional small amount of high-quality stories using an open LLM capable in Arabic. We chose Command R+ (104B) for the task as it is the more capable model in Cohere's family of multilingual models which scores highly on Huggingface's Open Arabic LLM Leaderboard (Elfilali et al., 2024). This new high quality content corresponds to 1% of the original noisy training data. Instead of deploying Command R+ locally (high compute requirement of 3× A100-80GB GPUs for inference), we generate the 20K stories via paid API calls for a cost of $100. We populated the prompts with samples from a set of random Arabic verbs, nouns and adjectives suited for 3-year-old stories, in addition to a selection story features as used in the original dataset. More details of the process can be found in Appendix A.2.

3.1.2 Continual pre-training

We further pre-train the 33M-ar SLM on the high-quality synthetically-generated data from the optimizer state saved after a single full epoch of training on the translated data. The additional data represent about 1% of the original training data. Therefore, the additional training time is small. The loss curves for the continual pre-training could be seen in Appendix B.3.The benchmarking results after refinement can be seen in Table 5. Continual pre-training shows performance improvement over the three metrics. Other refinement strategy such as multi-epochs, mixing with translation data in pre-training and then further pre-training with high-quality data will be explored in future work.

3.2 Interpretability using Sparse Auto-Encoders

In order to investigate whether the refinement process has managed to deal with the identified issues, we turn towards mechanistic interpretability. The traditional neuron-based methods have been shown recently not to be sufficient as neurons in language models appear to be *polysemantic* (Bricken et al., 2023), i.e., a single neuron might activate on multiple unrelated concepts, and traditional interpretability approaches cannot disentangle the real cause of a certain phenomenon. Recently, Dictionary Learning (Elhage et al., 2022a) has been developed as an alternative tool to interpret language models. The use of Sparse Auto-Encoders (SAE) (Bricken et al., 2023; Rajamanoharan et al., 2024; Makelov et al., 2024) to project neuron activations to a sparse and large dimensional *feature* space has led to a highly-interpretable representation that captures knowledge in generative language models not seen at neuron levels.

In order to understand the effect of our continual pre-training, we train a Sparse Auto-Encoder for both base and continually-trained models on the output of the last MLP layer. For these analyses we choose 2L-33M-ar model[2] (see Table 3).

3.2.1 Training SAEs

The architecture of our SAE is one hidden layer MLP trained as an autoencoder, with input weights as an encoder and output weights as the decoder. We chose an expansion factor of 16. The training data for SAEs is prepared as follow: For each context in the training data, the MLP activation vectors are collected after the SILU non-linearity for each token in the context. Activation vectors are sampled from 128 tokens within each context. These sampled vectors are then shuffled together, ensuring that the samples in a given batch originate from a variety of different contexts. Adam optimizer is used to minimize an objective function consisting of two components: the mean squared error loss and an L1 regularization term. The L1 regularization penalty is applied to the activations of the hidden layer within the autoencoder architecture. For a given feature the logit weight is defined as the product of a feature direction and the unembed $W_U W_{dec}[feature]$. The logit weight measures the direct effect of the features on the likelihood of next-token prediction. Examples of logit weight distribution can be found in (Johnny Lin, 2024)[3].

3.2.2 Token Set Enrichment Analysis

The Token Set Enrichment Analysis (TSEA) introduced in (Joseph Bloom, 2024) is borrowed from Gene Set Enrichment Analysis in bioinformatics (Subramanian et al., 2005) to statistically quantify if a set of features is strongly associated with a hypothesis. TSEA examines the relationship between the distribution of logit weights and predefined sets of tokens. The latter could represent various semantic or linguistic categories such as set of token of boy or girl names. TSAE would in that case identify which features inhibit or promote these sets of names. The steps in TSEA are: 1. Generate a library of token sets for the hypothesis under investigation. 2. Compute the *enrichment scores* for all features across the sets. The enrichment scores are running sum statistics that expose which features promote or suppress tokens in the hypothesis sets. 3. Identify elevated points in the enrichment scores. 4. Inspect features with high enrichment scores to validate the hypothesis.

3.2.3 Analyzing MT issues with TSEA

In this work, we investigate two hypotheses as case studies of the effect of continual pre-training with high-quality data: 1. The cultural bias of carrying English names to translated data. 2. The English language dialog follows the template “`[Quoted text],” said [Person]`, while in Arabic, the template is `Said [Person]: “[Quoted text]”`.

For cultural bias, we defined two token sets of 600 common English and Arabic first names. TSEA help identifying which set of names is higher represented in the model. Figure 2 shows the scatter plot of enrichment scores for English vs. Arabic names. Features above the diagonal activate stronger on English than Arabic names, and vice versa. We identified and marked the features farther from the diagonal with at least 2

[2] This model has previously been used for SAE analysis on English TinyStories: `https://huggingface.co/collections/lovish/tinystories-sae-regularization-comparison-65edb87a85cb3a3386c4b062`

[3] `https://www.neuronpedia.org/gpt2-small/8-res-jb/6649`

point difference between the English and Arabic name enrichment scores, indication strong bias. The plot shows a higher concentration of features representing English names, while the similar plot after continual pre-training in Figure 3 shows only one feature has the gap. The overall distribution is corrected away from English names, as also could be seen in generated samples. This illustrates the benefit of continual pre-training in correcting cultural bias.

The second case study is on the linguistic issue related to inappropriate dialog tagging in Arabic, which we noticed to be frequent in the translated TinyStories due to the quality of NLLB-3B. We defined the token set with English and Arabic persons names ending a sentence, and performed TSEA analysis before and after continual pre-training. Figure 5 shows the Manhattan plots of enrichment scores of both case studies. The first panel on the left shows the scores for dialog tagging for the model trained on translation data (a) and after continual pre-training (b). We carefully inspected the dashboards of top-3 features (2322, 3353, 14589) and (50, 1455, 9578). We concluded that the top-features in the updated model activate on correct dialog tag phrases whereas the base model shows clearly the issue.

Feature #3144 is a good example to illustrate both case studies. Figure 5 shows the feature dashboard[4]. The histogram on the top right is of sampled nonzero activations. The list of token of positive and negative logits underneath are the lowest and highest logit difference tokens of that feature, i.e., tokens whose probabilities of being sampled decrease or increase the most when the feature activates. Notice that the highest ranked tokens are of Arabic names, meaning that the model favors to produce an Arabic name after the verb

تدعى

(which means "called") that activates the feature highly as seen in the top activation panel to the left. The color highlights activation level of the token. A blue underline represents a lower loss (better prediction of that token), whereas a red underline represents a higher loss. The top activation examples illustrate correct examples of proper names ending a sentence.

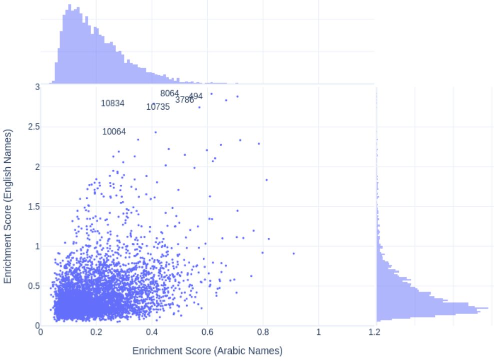


Figure 2: Scatter plot of feature enrichment scores for Arabic and English names in 2L-33M-ar.

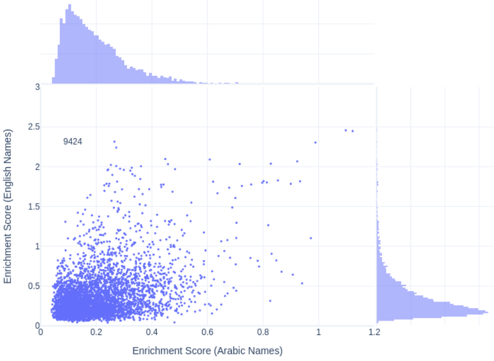


Figure 3: Scatter plot of feature enrichment scores for Arabic and English names for 2L-33M-ar-CP model after further pre-training.

4 Related Work

4.1 Continual pre-training

Continual pre-training has been previously used for domain adaption in LMs (Gupta et al., 2023; Ke et al., 2023). For example, general-purpose

[4] A guide to read the dashboard could be found in `https://transformer-circuits.pub/2023/monosemantic-features/index.html#setup-interface` and more details can be found in (Bricken et al., 2023; Bloom, 2024)

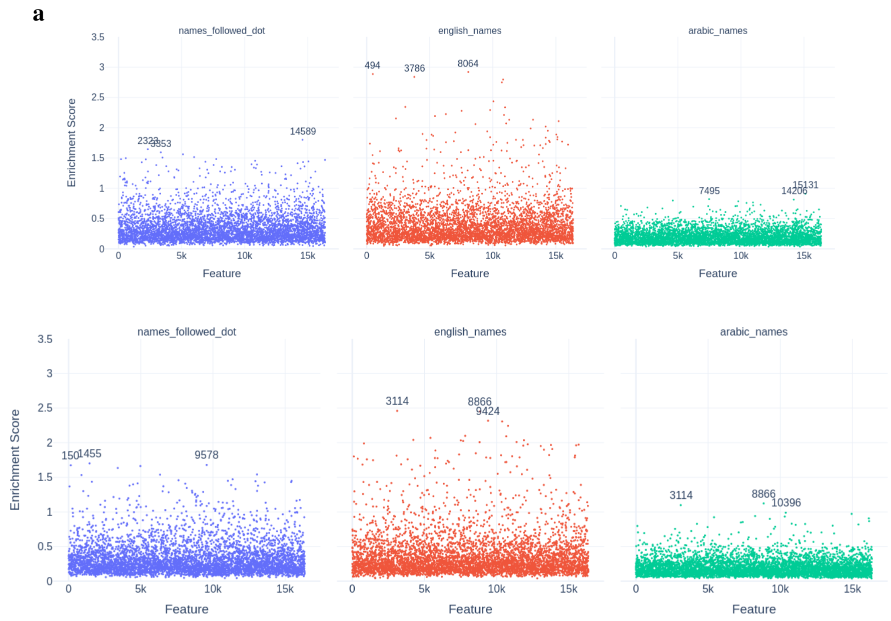
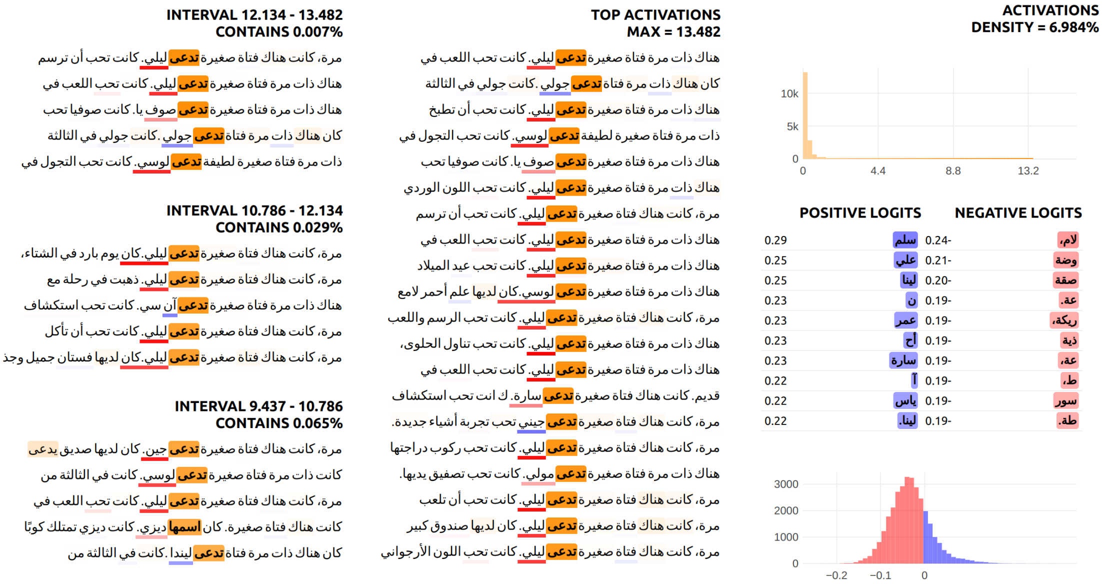


Figure 4: Manhattan plots of Enrichment Scores in (a) base and (b) further pre-trained models.



INTERVAL 12.134 - 13.482
CONTAINS 0.007%

مرة، كانت هناك فتاة صغيرة تدعى ليلي. كانت تحب أن ترسم
هناك ذات مرة فتاة صغيرة تدعى ليلي. كانت تحب اللعب في
هناك ذات مرة فتاة صغيرة تدعى صوف يا. كانت صوفيا تحب
كان هناك ذات مرة فتاة تدعى جولي .كانت جولي في الثالثة
ذات مرة فتاة صغيرة لطيفة تدعى لوسي. كانت تحب التجول في

INTERVAL 10.786 - 12.134
CONTAINS 0.029%

مرة، كانت هناك فتاة صغيرة تدعى ليلي.كان يوم بارد في الشتاء،
مرة، كانت هناك فتاة صغيرة تدعى ليلي. ذهبت في رحلة مع
هناك ذات مرة فتاة صغيرة تدعى آن سي. كانت تحب استكشاف
مرة، كانت هناك فتاة صغيرة تدعى ليلي. كانت تحب أن تأكل
مرة، كانت هناك فتاة صغيرة تدعى ليلي.كان لديها فستان جميل وجذ

INTERVAL 9.437 - 10.786
CONTAINS 0.065%

مرة، كانت هناك فتاة صغيرة تدعى جين. كان لديها صديق يدعى
كانت ذات مرة فتاة صغيرة تدعى لوسي. كانت في الثالثة من
مرة، كانت هناك فتاة صغيرة تدعى ليلي. كانت تحب اللعب في
كانت هناك فتاة صغيرة. كان اسمها ديزي. كانت ديزي تمتلك كوبًا
كان هناك ذات مرة فتاة تدعى ليندا .كانت في الثالثة من

TOP ACTIVATIONS
MAX = 13.482

هناك ذات مرة فتاة صغيرة تدعى ليلي. كانت تحب اللعب في
كان هناك ذات مرة فتاة تدعى جولي .كانت جولي في الثالثة
هناك ذات مرة فتاة صغيرة تدعى ليلي. كانت تحب أن تطبخ
ذات مرة فتاة صغيرة لطيفة تدعى لوسي. كانت تحب التجول في
هناك ذات مرة فتاة صغيرة تدعى صوف يا. كانت صوفيا تحب
هناك ذات مرة فتاة صغيرة تدعى ليلي. كانت تحب اللون الوردي
مرة، كانت هناك فتاة صغيرة تدعى ليلي. كانت تحب أن ترسم
هناك ذات مرة فتاة صغيرة تدعى ليلي. كانت تحب اللعب في
هناك ذات مرة فتاة صغيرة تدعى ليلي. كانت تحب عيد الميلاد
هناك ذات مرة فتاة صغيرة تدعى لوسي.كان لديها علم أحمر لامع
مرة، كانت هناك فتاة صغيرة تدعى ليلي. كانت تحب الرسم واللعب
هناك ذات مرة فتاة صغيرة تدعى ليلي. كانت تحب تناول الحلوى،
هناك ذات مرة فتاة صغيرة تدعى ليلي. كانت تحب اللعب في
قديم. كانت هناك فتاة صغيرة تدعى سارة. ك انت تحب استكشاف
مرة، كانت هناك فتاة صغيرة تدعى جيني تحب تجربة أشياء جديدة.
مرة، كانت هناك فتاة صغيرة تدعى ليلي. كانت تحب ركوب دراجتها
هناك ذات مرة فتاة صغيرة تدعى مولي. كانت تحب تصفيق يديها.
مرة، كانت هناك فتاة صغيرة تدعى ليلي. كانت تحب أن تلعب
مرة، كانت هناك فتاة صغيرة تدعى ليلي. كان لديها صندوق كبير
مرة، كانت هناك فتاة صغيرة تدعى ليلي. كانت تحب اللون الأرجواني

ACTIVATIONS
DENSITY = 6.984%

POSITIVE LOGITS		NEGATIVE LOGITS	
0.29	سلم	0.24-	لام،
0.25	علي	0.21-	وضة
0.25	لينا	0.20-	صقة
0.23	ن	0.19-	عة.
0.23	عمر	0.19-	ريكة،
0.23	أح	0.19-	ذية
0.23	سارة	0.19-	عة،
0.22	آ	0.19-	ط،
0.22	ياس	0.19-	سور
0.22	لينا.	0.19-	طة.

Figure 5: Dashboard of feature #3114 from SAEs trained on the last MLP layer of continually trained model 2L-33M-ar-CP. The Feature #3114 shows that cultural bias was corrected after continual pre-training.

LLMs are continually pre-trained for domain specific applications such as in financial applications (Xie et al., 2023) or new languages such as in AceGPT and SeaLLMs (Huang et al., 2023; Nguyen et al., 2023).

4.2 Mechanistic Intepretability

Despite the progress in training capable language models, underlying working mechanisms are still not well understood. This presents risks and challenges in terms of model safety and robustness, as the model may produce inconsistent, irrelevant, or harmful outputs when faced with unfamiliar or adversarial inputs. The ability to quantify and understand working mechanisms in language models at different phases of model development is an important research topic. The field of mechanistic interpretability aims at providing principled tools and methods to reverse engineer working mechanisms in language models (Elhage et al., 2021, 2022b).

5 Conclusion

In this work, we investigate the role of machine translated data in training language models for story generation in Arabic. We identify linguistic and cultural bias issues in the translated data. We propose to address these issues by further pre-training the models with a small amount of high-quality synthetic data. We investigate the effects of this intervention via Dictionary Learning tools. The trained Sparse Auto-Encoders shows a shift in the learned features towards corrected linguistic properties and reduced cultural bias.

6 Limitations

6.1 Limitations of GPT-Eval

The use of LLM-as-judge approach to evaluate open-ended generation comes with some limitations. The complexity of the task of evaluating consistency, grammar and creativity is sometimes challenging for GPT-4. We noticed its limitations in complex Arabic evaluation compared to English. A reassessment of GPT-4 in Arabic with respect to newer models such as Llama-3 and Command R+ would be helpful to decide on the best LLM judge.

6.2 Generalization of the Results to Larger Datasets

While TinyStories provides a testbed for exploring different facets of language models, the extension of this work to larger models faces challenges: as the model size grows, there is a need for a large dataset which shifts the burden on synthesizing diverse and high-quality dataset. The work done in models such as Phi-2 and Phi-3 are promising direction to address this challenge (Abdin et al., 2024; Li et al., 2023).

6.3 Extension to Other Domains

Developing small generative language models that exhibits emergent properties is challenging for tasks beyond creative writing. For example, applications related to generic question answering would require large amount of data and larger models. Further research is needed to identify interesting applications and define the requirements in terms of data volume and quality to train highly capable small language models for other applications.

6.4 Instruction Fine-Tuning

This work is limited to the training of base models. An instruction fine-tuning dataset of English TinyStories is available and similar work in the paper could be applied for the analysis of fine-tuned models. We will investigate this in future work.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Joseph Bloom. 2024. Open source sparse autoencoders for all residual stream layers of gpt2 small.

https://www.alignmentforum.org/posts/f9EgfLSurAiqRJySD/open-source-sparse-autoencoders-for-all-residual-stream.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, page 2.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ernie Chang, Hui-Syuan Yeh, and Vera Demberg. 2021. Does the order of training samples matter? improving neural data-to-text generation with curriculum learning. *arXiv preprint arXiv:2102.03554*.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.

Ali Elfilali, Hamza Alobeidli, Clémentine Fourrier, Basma El Amel Boussaha, Ruxandra Cojocaru, Nathan Habib, and Hakim Hacid. 2024. Open arabic llm leaderboard. https://huggingface.co/spaces/OALL/Open-Arabic-LLM-Leaderboard.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022a. Toy models of superposition. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/toy_model/index.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. 2022b. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1:1.

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Octopus: A multitask model and toolkit for Arabic natural language generation. In *Proceedings of ArabicNLP 2023*, pages 232–243, Singapore (Hybrid). Association for Computational Linguistics.

Kshitij Gupta, Benjamin Thérien, Adam Ibrahim, Mats L Richter, Quentin Anthony, Eugene Belilovsky, Irina Rish, and Timothée Lesort. 2023. Continual pre-training of large language models: How to (re) warm your model? *arXiv preprint arXiv:2308.04014*.

Oskar Holmström, Jenny Kunz, and Marco Kuhlmann. 2023. Bridging the resource gap: Exploring the efficacy of English and multilingual LLMs for Swedish. In *Proceedings of the Second Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL-2023)*, pages 92–110, Tórshavn, the Faroe Islands. Association for Computational Linguistics.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Ziche Liu, et al. 2023. Acegpt, localizing large language models in arabic. *arXiv preprint arXiv:2309.12053*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Joseph Bloom Johnny Lin. 2024. Announcing neuronpedia: Platform for accelerating research into sparse autoencoders. https://www.lesswrong.com/posts/BaEQoxHhWPrkinmxd/announcing-neuronpedia-as-a-platform-to-accelerate-research.

Johnny Lin Joseph Bloom. 2024. Understanding sae features with the logit lens. https://www.lesswrong.com/posts/qykrYY6rXXM7EEs8Q/understanding-sae-features-with-the-logit-lens.

Moussa Kamal Eddine, Nadi Tomeh, Nizar Habash, Joseph Le Roux, and Michalis Vazirgiannis. 2022. AraBART: a pretrained Arabic sequence-to-sequence model for abstractive summarization. In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 31–42, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual pre-training of language models. *arXiv preprint arXiv:2302.03241*.

Anis Koubaa, Adel Ammar, Lahouari Ghouti, Omar Najar, and Serry Sibaee. 2024. Arabiangpt: Native arabic gpt-based large language model. *Preprint*, arXiv:2402.15313.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2024. Madlad-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems*, 36.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. 2021. Pay attention to mlps. *Advances in neural information processing systems*, 34:9204–9215.

Aleksandar Makelov, George Lange, and Neel Nanda. 2024. Towards principled evaluations of sparse autoencoders for interpretability and control. *arXiv preprint arXiv:2405.08366*.

Sachin Mehta, Mohammad Sekhavat, Qingqing Cao, Max Horton, Yanzi Jin, Frank Sun, Iman Mirzadeh, Mahyar Najibikohnehshahri, Dmitry Belenko, Peter Zatloukal, and Mohammad Rastegari. 2024. Openelm: An efficient language model family with open training and inference framework.

Xuan-Phi Nguyen, Wenxuan Zhang, Xin Li, Mahani Aljunied, Qingyu Tan, Liying Cheng, Guanzheng Chen, Yue Deng, Sen Yang, Chaoqun Liu, et al. 2023. Seallms–large language models for southeast asia. *arXiv preprint arXiv:2312.00738*.

Guilherme Penedo, Hynek Kydlíček, Leandro von Werra, and Thomas Wolf. 2024. Fineweb.

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

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, Janos Kramar, Rohin Shah, and Neel Nanda. 2024. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2024. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 36.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, Alham Fikri Aji, Zhiqiang Shen, Zhengzhong Liu, Natalia Vassilieva, Joel Hestness, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Hector Xuguang Ren, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *Preprint*, arXiv:2308.16149.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*.

Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. 2005. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550.

Brian Thompson, Mehak Preet Dhaliwal, Peter Frisch, Tobias Domhan, and Marcello Federico. 2024. A shocking amount of the web is machine translated: Insights from multi-way parallelism. *Preprint*, arXiv:2401.05749.

Jörg Tiedemann, Mikko Aulamo, Daria Bakshandaeva, Michele Boggia, Stig-Arne Grönroos, Tommi Nieminen, Alessandro Raganato, Yves Scherrer, Raul Vazquez, and Sami Virpioja. 2023. Democratizing neural machine translation with opus-mt. *Language Resources and Evaluation*, pages 1–43.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson

Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, ..., and Thomas Wolf. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Yong Xie, Karan Aggarwal, and Aitzaz Ahmad. 2023. Efficient continual pre-training for building domain specific large language models. *arXiv preprint arXiv:2311.08545*.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.

Chen Zhang, Dawei Song, Zheyu Ye, and Yan Gao. 2023. Towards the law of capacity gap in distilling language models. *arXiv preprint arXiv:2311.07052*.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 73–81, Florence, Italy. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Tianyi Zhou, Shengjie Wang, and Jeffrey Bilmes. 2020. Curriculum learning by dynamic instance hardness. *Advances in Neural Information Processing Systems*, 33:8602–8613.

A Story Samples

A.1 Example NLLB-3B translation

Here is an example of a story and its translated version using NLLB-3B.

A.1.1 Original story in English

One day, a little boy named Tim went to the park.He saw a big tree and wanted to climb it.Tim was frightened, but he took a deep breath and started to climb. As he went up, he heard a small voice.The small voice said, "Please help me!"Tim looked around and saw a little bird.The bird had a hurt wing and could not fly.The bird was frightened too.Tim wanted to help the bird, so he thought of a plan.Tim took a big leaf from the tree and folded it.He made a soft bed for the bird. He put the bird on the leaf and climbed down.The bird was so happy and said, "Thank you, Tim!"The little boy and the bird became good friends.

A.1.2 Translated story using NLLB-3B

في يوم من الأيام ذهب صبي صغير يدعى تيم إلى الحديقة. رأى شجرة كبيرة وأراد أن يتسلقها. خاف تيم ، ولكنه أخذ نفسا عميقا وبدأ في التسلق. عندما صعد، سمع صوتًا صغيراً. قال الصوت الصغيراً: «من فضلك ساعدني» نظر تيم حوله ورأى طيرًا صغيرًا. كان للطائر جناح مجروحًا ولم يستطع الطيران. كان الطير خائفًا أيضًا. أراد تيم مساعدة الطائر، لذا فكّر في خطة. أخذ تيم ورقة كبيرة من الشجرة وطويها. صنع للطائر سريرًا ناعمًا. وضع الطائر على الورقة و تسلق للأسفل. كان الطائر سعيد جدا وقال: «شكرا لك، تيم !» أصبح الصبي الصغير والطائر أصدقاء جيدين.

A.2 Synthetic generation using Command R+

We prompt Command R+ to synthesize a small set of high-quality Arabic stories as describe in 3.1.1. The prompt we used is shown in Figure A.1, and the story generation features in Table 6.

Here is an example of a generated Arabic story:

كان هناك طفل صغير اسمه أحمد. أحمد كان مستعجل دائما ويحب اللعب في الخارج تحت الشمس. في أحد الأيام، بينما كان أحمد يلعب، بدأت السماء تمطر فجأة. أحمد لم يحب المطر أبدا لأنه كان يبلل ملابسه ويمنعه من اللعب. فكر أحمد في حل وقرر أن يصنع شمسية بنفسه حتى يستطيع اللعب في الخارج حتى عندما تمطر السماء.ذهب أحمد إلى أمه وسألها: «أمى، هل يمكن أن تساعديني في صنع شمسية؟ أريد أن ألعب في الخارج ولكن المطر يزعجني.»ابتسمت أمه وقالت: «بالتأكيد يا أحمد. يمكننا استخدام بعض الأقمشة القديمة وعصا طويلة لصنع شمسية رائعة. لكن يجب أن تكون صبورا لأن صنع الشيء يأخذ وقتا.»أحمد لم يكن صبورا أبدا لكنه وافق لأن أراد شمسية جميلة. عمل أحمد مع أمه بجد لعدة أيام. قطعوا القماش ولصقوه معا وحولوا العصا الطويلة إلى مقبض للشمسية. أخيرا، انتهوا من صنع الشمسية وكانت رائعة!أحمد كان سعيدا جدا وخرج إلى الخارج ليجرب

شمسيته الجديدة. لكن، عندما فتحها، حدث شيء غير متوقع. الرياح بدأت تهب بقوة وأحمد لم يستطع السيطرة على الشمسية. طارت الشمسية من يده واختفت في السماء. أحمد بكى وحزن جدا لأنه فقد شمسيته الجميلة التي عمل بجد لصنعهاالأم جاءت لعنده وحاولت أن تواسيه: «لا تبكي يا أحمد. أنا فخورة بك لأنك عملت بجد وصنعت شيئًا رائعا. لكن تذكر، عندما نريد شيئًا ما، يجب أن نكون صبورين ونعمل بجد لتحقيقه. في بعض الأحيان، حتى إذا فعلنا كل شيء بشكل صحيح، قد لا نحصل على ما نريد. هذا جزء من الحياة ويجب أن نتعلم كيف نتعامل مع خيبات الأملأحمد تعلم درسا مهما ذلك اليوم. تعلم أن العمل الجاد والصبر مهمان، ولكن أيضا أن هناك أشياء خارجة عن سيطرتنا. في بعض الأحيان، حتى إذا احتجنا شيئًا ما بشدة، قد لا نحصل عليه. هذا لا يعني أن نتوقف عن المحاولة، ولكن أن نتعلم كيف نتأقلم ونستمر في النمو.

```
Write a short story (3-5 paragraphs)
in Arabic language which only uses
very simple words that a 3 year old
child would likely understand.  The
story should use the verb "{verb}",
the noun "{noun}" and the adjective
"{adjective}".  The story should have
the following features:  {features}.
Remember to write in Arabic and to
only use simple words!
```

Figure A.1: Prompt for synthesizing Arabic stories with Command R+, an Arabic-capable LLM.

A.3 Completion example

Here is an example of a prompt from the test set (top panel of Figure A.2) and model 33M-ar completion (bottom panel).

B Training Details

B.1 Process calibration using original TinyStories

We calibrate the training process by first training the chosen architectures on the original TinyStories dataset and comparing the benchmarks with the published results. Table 7 shows GPT-4 scores for some trained models, and Figure A.3 shows the loss curve for the training.

في يوم من الأيام، كانت الجدة في المطبخ تستعد للطهي. لديها مكونات لطبق شهي. قالت الجدة لحفيدتها: «هل ستساعديني في إعداد هذه الوجبة؟» ردت حفيدتها، جيني: «نعم، سأساعدك!» بدأت الجدة وجيني في تحضير المكونات. كانت جيني غير ماهرة. أسقطت ***

الدقيق في الفرن بينما كانوا يعدون العشاء على الموقد. كان رائحة المطبخ لذيذة. عندما كان العشاء جاهزاً ، وضعت الجدة وجيني وعاءً كبيرًا من الطعام اللذيذ على الطبق. بدا لذيذًا جدًا. أعطت الجدة جيني ملعقة كبيرة لتحريك الطعام...

Figure A.2: An example of a completion by the model 33M-ar. The top panel shows the prompt from the test set ending in an incomplete sentence followed by *** to assess the model's ability to complete a coherent transition sentence. The bottom panel shows the model continuation.

B.2 Pre-training loss curves

Figures A.4 and A.5 show the training and validation loss, respectively, for the pre-training stage using translated TinyStories.

B.3 Continual pre-training loss curves

Figures A.6 and A.7 show the training and validation loss, respectively, for the continual pre-training stage using the Arabic synthetic story dataset.

B.4 Feature dashboard examples

Figure A.8 is a dashboard summarizing the properties of the selected feature #14589. The top right histogram gives the activation distribution for the feature. The panel underneath shows the highest and lowest ranked tokens based on the logit difference when the feature is ablated. The text panels show training samples for different feature activation intervals. Feature #14589 captures the verb 'say' in Arabic in different tense

Feature	Prob.	Instruction
Dialogue	0.6	the story should contain at least one dialogue
BadEnding	0.3	the story has a bad ending
Conflict	0.1	the story has some form of conflict in it
Moral Value	0.1	the story has a moral value
Fore-shadowing	0.1	the narrative uses foreshadowing or setup and payoff
Twist	0.3	something unexpected happens / there is a plot twist

Table 6: For each story, a set of features is randomly selected according to the probability and definition described in the table.

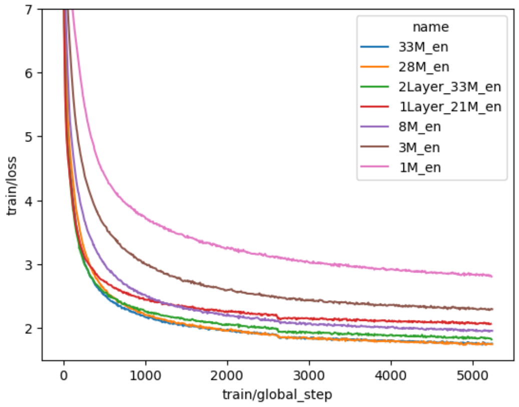


Figure A.3: Train loss of models trained on English TinyStories.

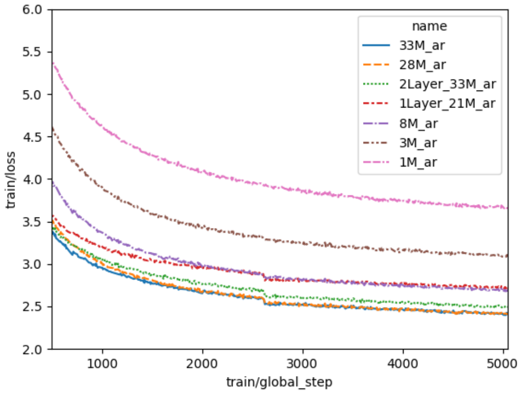


Figure A.4: Training loss of models trained on translated TinyStories.

model	Grammar	Creativity	Consistency
gpt2-large	5.52	1.31	1.68
33M-en	6.40	6.63	6.97
Mistral-7b	7.86	6.88	8.04

Table 7: Performance results of a model trained on the original TinyStories dataset against baselines (gpt2-large (Radford et al., 2019) and Mistral-7B (Jiang et al., 2023)) in order to calibrate the training process.

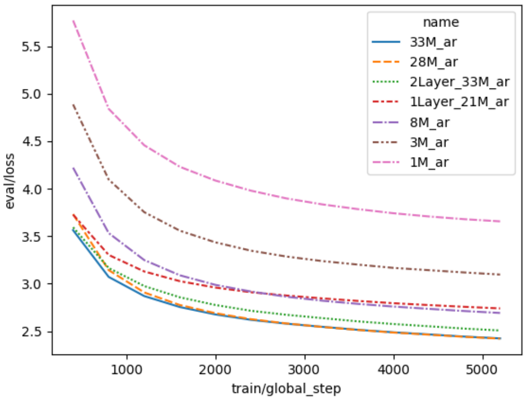


Figure A.5: Validation loss of models trained on translated TinyStories.

forms following a quote. This construction is not correct in Arabic as the use of the quote tagging is given before the quote. This reversed structure has been inherited from English due to the low-quality translation of the dataset. This feature helped us reveal some of the limitations of our model trained with low-quality translation data.

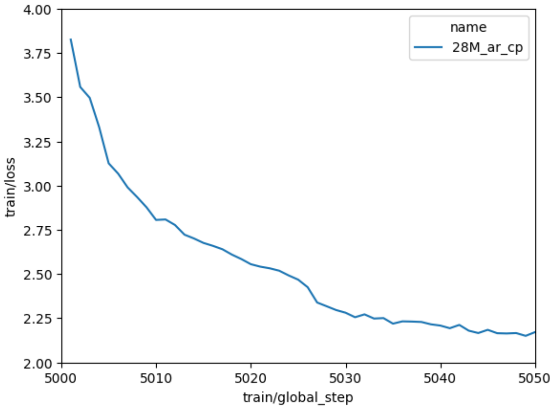


Figure A.6: Training loss during continual pre-training with high-quality synthesized data.

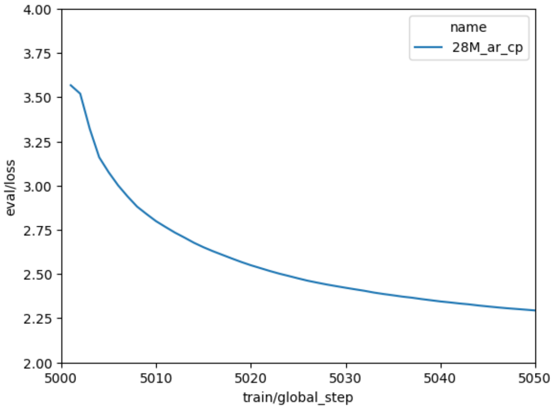
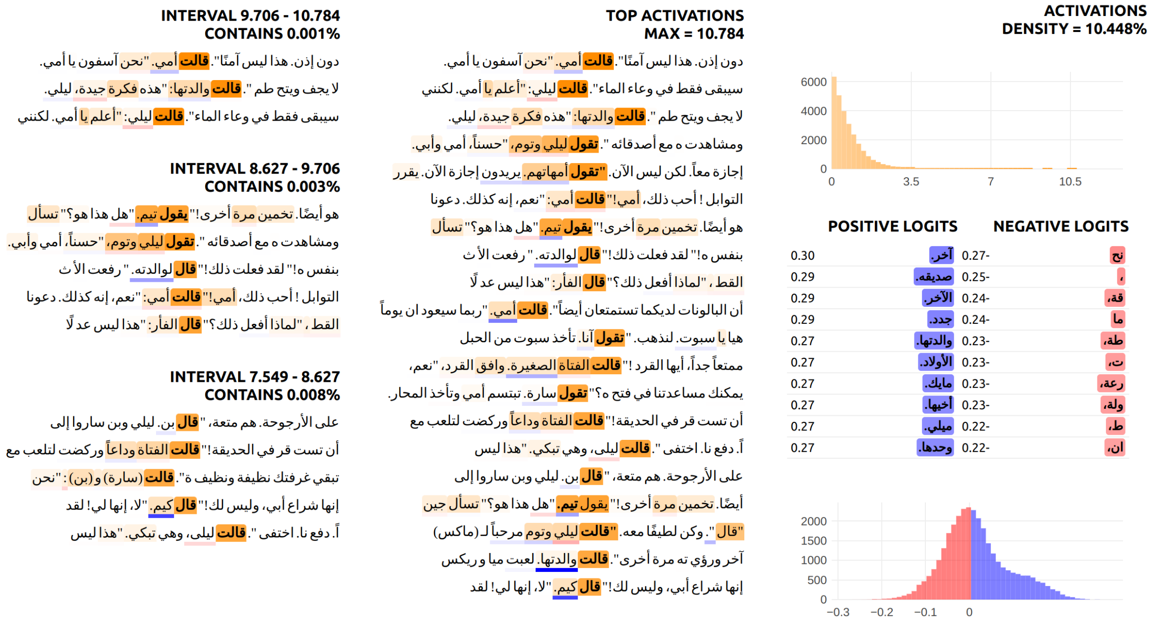


Figure A.7: Evaluation loss during continual pre-training with high-quality synthesized data.



INTERVAL 9.706 - 10.784
CONTAINS 0.001%

دون إذن. هذا ليس آمنًا". قالت أمي. "نحن آسفون يا أمي.

لا يجف ويتح طم ". قالت والدتها: "هذه فكرة جيدة، ليلي.

سيبقى فقط في وعاء الماء". قالت ليلي: "أعلم يا أمي. لكنني

INTERVAL 8.627 - 9.706
CONTAINS 0.003%

هو أيضًا. تخمين مرة أخرى!" يقول تيم. "هل هذا هو؟" تسأل

ومشاهدت ه مع أصدقائه ". تقول ليلي وتوم، "حسناً، أمي وأبي.

بنفس ه!" لقد فعلت ذلك!" قال لوالدته. " رفعت الأ ث

التوابل ! أحب ذلك، أمي!" قالت أمي: "نعم، إنه كذلك. دعونا

القط ، "لماذا أفعل ذلك؟" قال الفأر: "هذا ليس عد لًا

INTERVAL 7.549 - 8.627
CONTAINS 0.008%

على الأرجوحة. هم متعة، " قال بن. ليلي وبن ساروا إلى

أن تست قر في الحديقة!" قالت الفتاة وداعاً وركضت لتلعب مع

تبقي غرفتك نظيفة ونظيف ة". قالت (سارة) و(بن): "نحن

إنها شراع أبي، وليس لك!" قال كيم. "لا، إنها لي! لقد

اً. دفع نا. اختفى ". قالت ليلى، وهي تبكي. "هذا ليس

TOP ACTIVATIONS
MAX = 10.784

دون إذن. هذا ليس آمنًا". قالت أمي. "نحن آسفون يا أمي.

سيبقى فقط في وعاء الماء". قالت ليلي: "أعلم يا أمي. لكنني

لا يجف ويتح طم ". قالت والدتها: "هذه فكرة جيدة، ليلي.

ومشاهدت ه مع أصدقائه ". تقول ليلي وتوم، "حسناً، أمي وأبي.

إجازة معاً. لكن ليس الآن. "تقول أمهاتهم. يريدون إجازة الآن. يقرر

التوابل ! أحب ذلك، أمي!" قالت أمي: "نعم، إنه كذلك. دعونا

هو أيضًا. تخمين مرة أخرى!" يقول تيم. "هل هذا هو؟" تسأل

بنفس ه!" لقد فعلت ذلك!" قال لوالدته. " رفعت الأ ث

القط ، "لماذا أفعل ذلك؟" قال الفأر: "هذا ليس عد لًا

أن البالونات لديكما تستمتعان أيضاً". قالت أمي. "ربما سيعود ان يوماً

هيا يا سبوت. لنذهب. "تقول آنا. تأخذ سبوت من الحبل

ممتعاً جداً، أيها القرد !" قالت الفتاة الصغيرة. وافق القرد، "نعم،

يمكنك مساعدتنا في فتح ه؟" تقول سارة. تبتسم أمي وتأخذ المحار.

أن تست قر في الحديقة!" قالت الفتاة وداعاً وركضت لتلعب مع

اً. دفع نا. اختفى ". قالت ليلى، وهي تبكي. "هذا ليس

على الأرجوحة. هم متعة، " قال بن. ليلي وبن ساروا إلى

أيضًا. تخمين مرة أخرى!" يقول تيم. "هل هذا هو؟" تسأل جين

"قال ". وكن لطيفًا معه. "قالت ليلي وتوم مرحباً لـ(ماكس)

آخر ورؤي ته مرة أخرى". قالت والدتها. لعبت ميا و ريكس

إنها شراع أبي، وليس لك!" قال كيم. "لا، إنها لي! لقد

ACTIVATIONS
DENSITY = 10.448%

POSITIVE LOGITS		NEGATIVE LOGITS	
0.30	آخر.	0.27-	نح
0.29	صديقه.	0.25-	،
0.29	الآخر.	0.24-	قة،
0.29	جدد.	0.24-	ما
0.27	والدتها.	0.23-	طة،
0.27	الأولاد.	0.23-	ت،
0.27	مايك.	0.23-	رعة،
0.27	أخيها.	0.23-	ولة،
0.27	ميلي.	0.22-	ط،
0.27	وحدها.	0.22-	ان،

Figure A.8: Dashboard of feature #14589 from SAEs trained on the last MLP layer of the base model 2L-33M-ar. It corresponds to token set in TSEA formed as first name followed by punctuation. This feature captures the quote tagging issue discussed in Section 3.2.3.