ALLaM: A Series of Large Language Models for Arabic and English

Anonymous ACL submission

Abstract

In this work, we present ALLaM: **A**rabic **L**arge **La**nguage **M**odel, a series of large language models to support the ecosystem of Arabic Language Technologies (ALT). ALLaM is carefully trained, considering the values of *language alignment* and *transferability* of knowledge at *scale*. The models are based on an autoregressive decoder-only architecture and are pretrained on a mixture of Arabic and English texts. We illustrate how the second-language acquisition via vocabulary expansion can help steer a language model towards a new language without any major catastrophic forgetting in English. Furthermore, we highlight the effectiveness of using translation data and the process of knowledge encoding within the language model's latent space. Finally, we show that effective alignment with human preferences can significantly enhance the performance of a large language model (LLM) compared to less aligned models of a larger scale. ALLaM achieves state-of-the-art performance in various Arabic benchmarks, including MMLU Arabic, ACVA, and Arabic Exams. Our aligned models improve both in Arabic and English from its base aligned models.

1 Introduction

Language modeling has significantly progressed from its humble origins, transitioning from fundamental probabilistic methods to complex neural priors. The foundational work by Shannon (1951) on the information theory of language laid the groundwork for predicting the next word in a sequence, which was initially tackled by Bengio et al. (2003) in neural space. The field experienced a substantial leap with the introduction of LSTMs (Hochreiter and Schmidhuber, 1997) in language model (LM) (Peters et al., 2018), which could capture longer dependencies in LMs but lacked scaling capability. The emergence of scalable and distributed architectures like Transformers (Vaswani et al., 2017), the

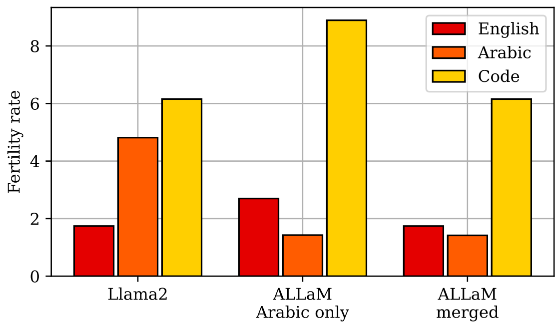


Figure 1: Comparison of Fertility Rates of LLaMa-2 and ALLaMtokenizers. The chart illustrates the fertility rates across three models: LLaMa-2, ALLaMArabic only, and ALLaM merged with LLaMa-2 tokenizer, with datasets in English, Arabic and Code.

potential for precisely (Kaplan et al., 2020; Hoffmann et al., 2022) compressing web-scale data has resonated in recent years with the advancements of *Generative Pretraining* (Radford et al., 2018; Brown et al., 2020; Anil et al., 2023).

With the release of `ChatGPT` (OpenAI, 2022), followed by the introduction of more frontier class models `Gemini` (Google, 2024), `Claude` (Anthropic, 2022), `Reka` (Ormazabal et al., 2024), `Mistral` (Mistral, 2024), `Llama-3` (Meta, 2024) and recently released `Qwen-2` (Alibaba, 2024), generative models have experienced a significant leap from previous models (Laskar et al., 2023), raising potential implications of Artificial General Intelligence (Hendrycks and Mazeika, 2022; Marcus, 2022). This advancement has spurred discussions across various fields, including ethics, economics, and technology (Weidinger et al., 2021). Judging from the initial capabilities (Bubeck et al., 2023), the potential of these frontier models are reinventing the way humans interact with machines, impacting social norms, productivity, trends, and culture on a broader scale (Zhou et al., 2024). However, most of these frontier-class models are primarily trained on English or a few languages and often lack integration of localized regional cultures

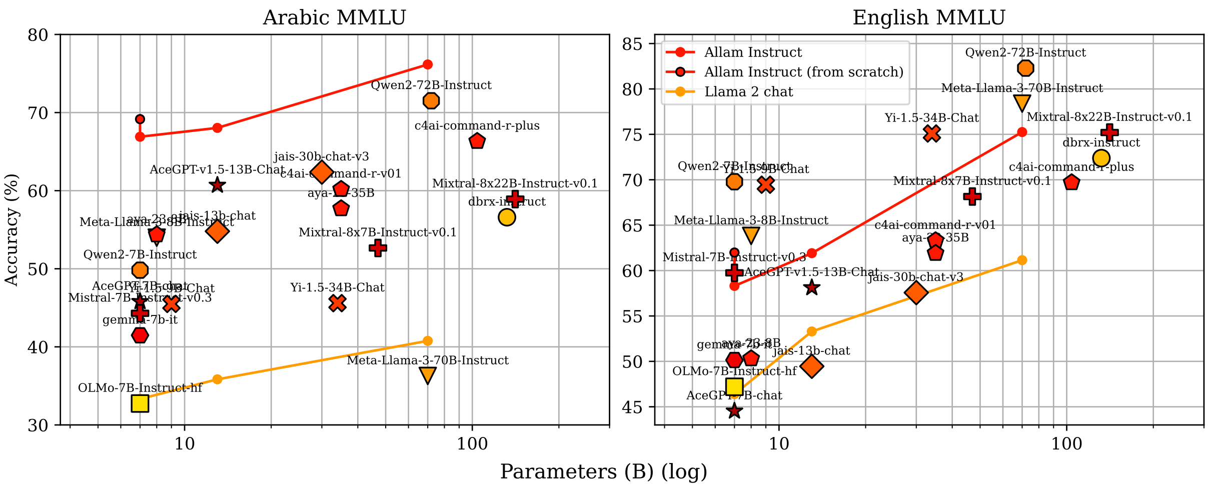


Figure 2: Performance of Various Models on Arabic (Koto et al., 2024) and English (Hendrycks et al., 2020) MMLU Benchmarks. `ALLaM` shows impressive improvement from it's base model `Llama 2`.

and norms (Naous et al., 2024), risking ***slow***, ***irreversible manipulation*** of regional identities and potentially leading to cultural homogenization.

The significant training costs of LLMs and their environmental impact have become major concerns in recent years (Strubell et al., 2019). The vast computational resources required to train LLMs contribute to substantial carbon emissions (Luccioni and Hernandez-Garcia, 2023). Governments [1] and non/for-profit organizations (Dodge et al., 2022; Google, 2021; Amazon, 2021), are increasingly aware of these issues. This awareness has led to discussions about the ethical implications of AI development and the need for sustainable practices concerning "*When and how to **scale** the training of these models*" To address these concerns, instead of scaling ***fast***, we have opted to continue training from a *well-documented*, *strong*, but potentially *under-trained* pre-trained model rather than starting from a randomly initialized model. We initialize our model from **Llama 2** (Touvron et al., 2023) weights. This approach offers several key advantages that align with both our technical goals and our commitment to sustainable practices.

Technically, *continue pre-training* a model in a new language can aid in understanding **Second Language Acquisition** (SLA) (Swain and Lapkin, 1995), popularized by Bari et al. (2020) in NLP and recently adopted by Nguyen et al. (2023). This process involves the challenging task of incorporating an additional distribution without *compromising the source*. For instance, if a pre-trained model was initially trained in English, expanding to an additional language presents the specific challenge of addressing tokenization issues. Figure 1 gives an overview of ALLaM tokenizers. We expand the vocabulary of Llama 2 tokenizer from an Arabic-only ALLaM tokenizer. With the vocabulary expanded model, we continue pre-train our model for additional 1.2 Trillion tokens [2] on English and Arabic data mixtures and show impressive improvement over the Llama 2 base model. Finally, we apply these learnings to pre-train and align a 7B parameter model from scratch [3], showing impressive improvements across the range of 7B parameter open models. In general, our contributions are listed below:

- We present the ALLaM model series, a collection of large language models developed specifically for Arabic and English languages, with the goal of supporting the cultural values of the *Arab World*. We train four models at three different scales: 7B, 13B, and 70B model initialized by Llama weights and a 7B model from scratch.
- Unlike recent trends, we explain our training methodologies and the thought process behind the decision-making involved in training the LLM. We provide necessary ablation studies for most of our crucial decisions.
- Our model achieves state-of-the-art results in Arabic as well as improving overall English performance of the original `LLaMa-2` model. Check figure 2 for a quick overview.

[1] `https://www.cnrs.fr/en/update/jean-zay-supercomputer-recycling-its-heat`

[2] For ALLaM-70B model, we train on 600B tokens

[3] Model initialized with random weights

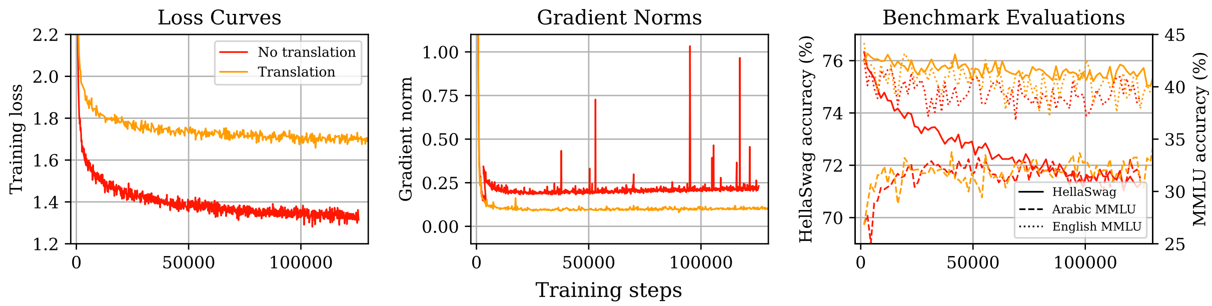


Figure 3: Effect of Arabic Translation Data in Pretraining.

2 Pretraining

Pretraining language models on trillions of natural language tokens represents the bulk of cost required to build an effective language model. This large investment of time and compute precludes experimentation or ablation for every decision. Thus, before starting to train ALLaM from random initialization, or "scratch", we experiment in the continue-pretraining regime. As the name implies, Continue pretraining is the practice of warm-starting a pretraining experiment from an already pretrained LM.

2.1 Pretraining Data

Starting from a Llama 2 pretrained model, we continue pretraining the ALLaM-7B and ALLaM-13B models on 1.2T tokens, covering both English and Arabic languages. For the ALLaM-70B model, we train on 600B tokens.We included English data in our mixture to avoid degrading the performance of our model on English. For English, we harnessed subsets from Dolma-v1 (Soldaini et al., 2024) and Pile (Gao et al., 2021) datasets e.g., Dolma CC, The Stack (Kocetkov et al., 2022) and PeS2o, and PubMed, DM-Math (Saxton et al., 2019) and Stack-Exchange (Soboleva et al., 2023).

Our Arabic pretraining data include inhouse crawled diverse sources covering Web documents, news articles, books (literature, religion, law and culture, among others), Wikipedia (over 1M articles), and audio transcripts (books and news)[4]. To ensure high quality Web data, we applied the following processing steps: *(i)* Drop documents with language identification score < 95%, *(ii)* Drop short documents that are less than 30 words, *(iii)* Drop documents with duplicate URLs, high ratio of spam and stop words, *(iv)* Drop duplicate documents (using exact matching; although we experimented with fuzzy matching but we found it to be harsh and given that the Arabic data is scarce we opted not to use fuzzy matching for this version).

[4]We are currently working on systematic auditing of our pretraining data. Right now we do not have any timeline or visibility when or if we can share our data for research.

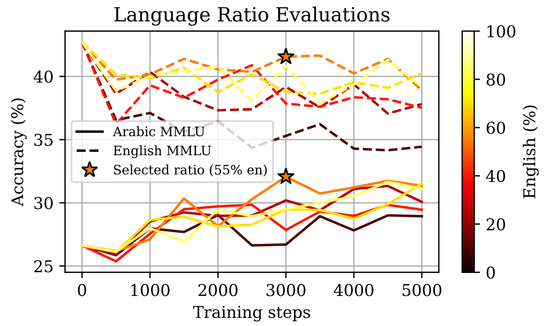


Figure 4: To find the right Arabic/English language mixture that acquires Arabic knowledge while still retaining English, we conducted an ablation over 20B tokens, in which We found that a 55/45 English/Arabic ratio achieves the best trend in performance, as measured via English and translated Arabic MMLU.

Additionally, we extended our Arabic data with translated English content using an in-house machine translation system. We translated the following English datasets from Dolma: Wikipedia, books, C4 and peS2o, which also are part of our English data, the hypothesis is that this will improve English-Arabic language alignment, leading to a better Arabic model. Figure 3 demonstrates the impact of Arabic translation dataset in the pretraining data mixture. While models trained without translation data exhibit lower training loss, those trained with translation data show more stable training, as evidenced by fewer spikes in gradient norms. Incorporating Arabic translation data in the pretraining dataset mitigates catastrophic forgetting in English. In total, we curate `500B` arabic tokens[5].

Data Mixture. To build a performant model in both English and Arabic, we conducted experiments to figure out an optimal language mix. Fig 4 gives an overview of data-mixture experiments on our curated English-Arabic corpus. We conducted the experiments with the same sampling ratio (Table 1) and data order. We observe best trend in performance with 55 : 45 English:Arabic data mix.

[5]Token counted by our merged tokenizer.

Domain	English	Arabic		Overall
		Natural	Translated	
Web	31%	71%	65%	48%
Books	9%	13%	12%	11%
Wiki	—	0.70%	0.61%	0.3%
News	—	14%	—	3%
Science	16%	—	22%	14%
Code	39%	—	—	21%
Math	5%	—	—	2.5%
Other	—	1.3%	0.39%	0.2%
Lang Mix	55%	22.5%	22.5%	100%
Tokens	660 B	270 B	270 B%	1200 B

Table 1: ALLaM Pretraining data mixture. We upsample data to match the mixture rates when needed. (Each column sums to 100%)

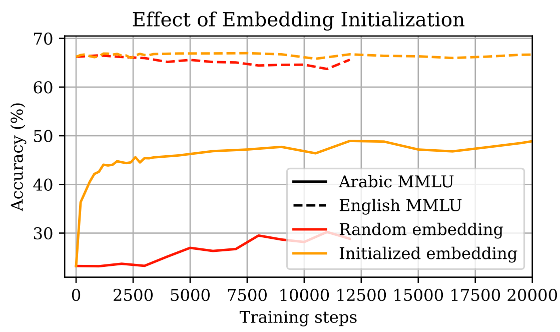


Figure 5: Effect of *Random initialization* vs *embedding initialization* during the start of continue pre-training.

Table 1 shows the language and category mixing distributions for English, Arabic Natural, Arabic Translated and final mix. As depicted, and following mainstream work, Web data constitutes the highest ratio with 71%, 65% and 48% of the AR Natural, AR Translated and Final, respectively. We limited the contribution of Web English data to 31%, as Llama 2 base model was trained on Web data already and increasing its ratio might degrade performance. We ensured that high-quality sources such as books, news articles and code are well-represented in our mixture.

2.2 Continued pretraining

Open-source and open weight models present an attractive option to conduct pretraining experiments cheaply, however, they also present challenges since most such models do not natively support Arabic or other languages. We develop a simple approach to enhance any language model with capabilities in new languages (e.g. language expansion). The approach relies on two steps: (i) tokenizer augmentation and (ii) expanded vocabulary learning. We demonstrate that this approach leads to minimal degradation of capabilities in the original language.

Tokenization To calculate the fertility of our tokenizers, we subsample the entire training corpus and use this subsample as test dataset.

Existing open-weight language models (e.g., Llama 2) tokenize Arabic (and other languages) poorly, often splitting words down to the character level or even relying on byte-fallback mechanisms for tokenization. This results in inefficient training, as the pretraining corpus size is inflated, and unoptimized inference, since the model must generate more tokens per word. Additionally, the context length is reduced because it is based on a fixed number of tokens. To address these issues, we use a corpus of text in the target language to train a tokenizer specialized in that language. We then merge the original tokenizer with the language-specific tokenizer. Merging is accomplished by adding all tokens from the language-specific tokenizer that do not exist in the original tokenizer. As shown in Figure 1, this effectively reduces the fertility rate in the target language of the merged tokenizer to the level of the language-specific tokenizer.

Newly added tokens in the merged tokenizer have no associated embedding representations in the pretrained language model's weights. To learn these representations, we experiment with two approaches: *(i)* random initialization and *(ii)* initialization from combined representations of tokens in the original tokenizer. Approach *(ii)* is accomplished by tokenizing the vocabulary of the new tokenizer using the original tokenizer. The associated representations of this tokenization are then averaged and assigned as the vector representation of the new token. Since we work with tokenizers with byte-fallback, such a tokenization is guaranteed to exist. Figure 5 provides an overview of our initialization method. Initializing the new embeddings from the combination of previously learned 2T token trained embeddings gives a significant boost to the learning of a new language. Figure 1 gives an overview of our tokenizers.

Learning rate In all of our continued pretraining experiments, we used the final learning rate of the pretrained language model (usually 3e-5). We experimented with approaches to gradually increase the learning rate and then decay it but found limited success. Such models typically exhibited catastrophic forgetting, indicated by significant drops in performance in the source language. We also considered optimizer state warmup (as open-weight models typically do not include the optimizer states) but found this had little effect on

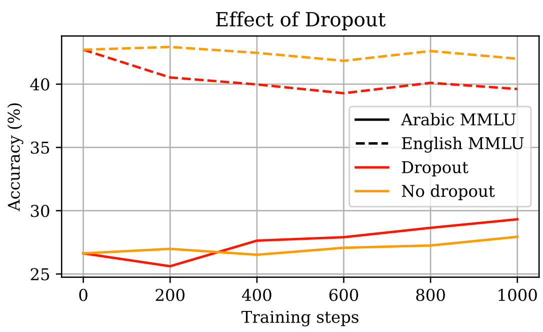


Figure 6: Effect of *Dropout* during the start of continue pre-training experiments.

performance. Figure 6 provides an overview of adding dropout during continued pretraining. We observe that adding dropout helps the Arabic language, as it acts as a regularizer for the new distribution. However, Llama 2 was pretrained on 2T tokens without any dropout, and adding dropout negatively impacts the source distribution. Considering this trade-off, we decided not to add dropout in the continue pretraining stage.

2.3 Pretraining from scratch

We were able to curate `500B` Arabic tokens. Following (Hoffmann et al., 2022; Touvron et al., 2023), training a high-quality English model from scratch requires a substantial amount of tokens. Even when pretraining from random initialization, we find it beneficial to start training with a high-resource language (en) and then continue pretraining to Arabic.

In pre-training from scratch, selecting and identifying good training dynamics requires spending a lot of tokens, as different evaluations start to discriminate at different stages [6]. This may require extensive ablation studies to determine the optimal setup. Our initial experiments with 1B parameter models show that training with two languages can sometimes degrade the performance in English or result in slow learning of both language distributions. We also hypothesize that low-resource languages can dilute in the large volume of high-resource language data, even with careful tuning.

On the contrary, our continued pretraining from scratch receipe retains the natural English distribution without catastrophic forgetting, effectively transferring knowledge from one distribution to another. Judging by this trade-off, we decide to first achieve a good English distribution before applying the same approach for large-scale language alignment. The only difference here is that there is no need for vocabulary expansion.

[6]For example, a 7 billion parameter model begins to show discrepancies in MMLU at the 1 trillion tokens range.

3 Alignment

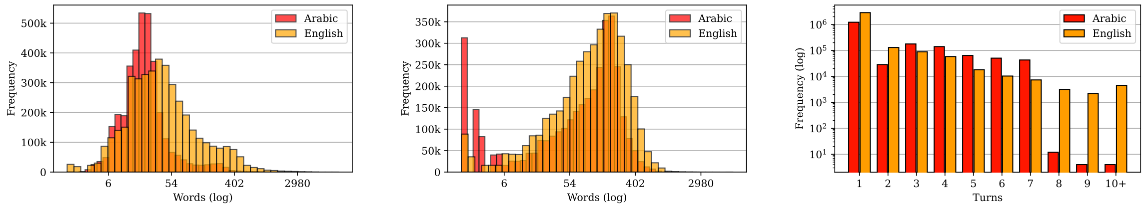
Building effective LLMs requires ensuring they perform well and adhere to ethical standards and user expectations. This alignment process is crucial, especially for models used in diverse linguistic and cultural contexts.

Supervised Finetuning (Section 3.1) refines a pre-trained model using a carefully selected dataset relevant to specific tasks and domains. Preference training (Section 3.2), on the other hand, aligns the model's outputs with human values and preferences by prioritizing responses that meet user expectations and ethical guidelines. Together, these methods create reliable and ethically sound LLMs for real-world use.

3.1 Supervised Finetuning Training

Data. Our Supervised Finetuning (SFT) data is curated from a diverse array of sources. For English, we primarily use public web content as our main source, offering a broad range of high-quality and especially diverse prompts. In contrast, our Arabic data comes from a combination of public and proprietary sources to ensure comprehensive coverage and relevance. We utilize classifiers, human and/or generative models (Ding et al., 2023) to *identify/interact* if the text can be considered suitable for supervised finetuning and/or if we can generate an SFT dataset from any context. To gather data from the source, we collect seed websites or data sources, which involves utilizing domain experts, prompt librarians, local institutes specializing in areas such as Arabic language, history, and politics, the use of permissible commercial LLMs to generate data, and machine translation models to convert rich English SFT data into Arabic. Our datasets cover various domains and capabilities, ensuring the model's proficiency in handling tasks across education, history, Arabic linguistics, politics, religion, computer science, and other fields. The entire collection is named as `Ultra-Instinct`, which is not *human generated* rather *human driven*.

Quality Is All You Need. Unlike Zhou et al. (2023); AI et al. (2024) we hypothized that scaling SFT data can unlock diverse capability as well as improve responsiveness to the prompts. Initially we crawled public web for supervised finetuned samples. The first version (v1) of `Ultra-Instinct` includes 6M samples each from English and Arabic, while the second version (v2), is a reduced version



(a) Prompt word count. (b) Response word count. (c) Number of conversation turns.

Figure 7: SFT data distributions.

Quality Metric	V1		V2	
	Prompt	Response	Prompt	Response
Word length	146.94	97.19	60.81	136.47
Lexical diversity	76.34	75.25	85.29	69.53

Table 2: Comparison of average word length and lexical diversity for (v1) and (v2) in prompts and responses.

with half the number of samples. For v1, we did not implement rigorous quality checks or extensive data removal. In contrast, v2 underwent strict quality checks and random human assessments. Our quality checks for v2 included *(i)* Assessments based on instruction/response word length, *(ii)* Lexical and semantic diversity, exact and near-exact lexical deduplication, *(iii)* The removal of low quality machine-translated Arabic data from English sources, and ensuring diversity in questions and commands. For detailed metrics on instruction and response lengths and lexical diversity, see Table 2.

Figure 7a and Section 3.1 shows the distribution of the prompts and responses in v2, respectively.

Version	MMLU			Exams (ar)	ACVA	ETEC
	Huang et al. (2023)	Koto et al. (2024)	en			
`Ultra-Instinct v1`	51.0	68.0	63.8	56.8	79.8	66.8
`Ultra-Instinct v2`	51.39	68.49	63.3	56.8	76.66	65.91

Table 3: Comparative results of `Ultra Instinct` versions, v1 and v2, across various evaluation datasets.

To extrinsically evaluate the impact of higher quality SFT data, we trained two 13B models using v1 and v2 datasets. Despite v2 containing 50% less data, both versions performed equally well on English and Arabic evaluation benchmarks. This reduction in data volume led to faster training times and reduced costs without compromising performance. Table 3 provides a detailed comparison of the 13B model results on `Ultra-Instinct` v1 and v2. `Ultra-Instinct` contains a large amount of multi-turn conversations. Figure 7c shows the distribution of "`# of turn`" from `Ultra-Instinct`.

While training the SFT model, we encountered an issue with the tokenizer. Llama 2 tokenizer was trained using `sentencepiece`[7], which breaks the beginning and end of sequence token with multiple tokens, adversely affecting long multi-turn conversations. To address this issue, we patched `sentencepiece` using the `huggingface` `LlamaTokenizer` wrapper. During many stages of training we saw that having 1% of noisy text (i.e., empty response) can visibly affect the model.

[7] `https://github.com/google/sentencepiece`

3.2 Preference Training

After SFT, models are able to converse in multi-turn conversations. However, they are not fully aligned with human preferences. For example, our SFT models were terse and had limited guardrails. To circumvent these issues, we performed preference tuning with human verified samples via Direct Preference Optimization (DPO) (Rafailov et al., 2024).

The DPO inputs we utilized were sourced from early model testers and a manually curated selection of domains, such as questions related to ethics or model ownership. DPO training necessitates both negative and positive output samples to train a reward model. We relied on the testers' feedback to identify the positive outputs. In the absence of positive outputs, we generated and verified positively aligned outputs. While (Tunstall et al., 2023) utlized preference data from AI Feedback (AIF) at *scale*, we adopt a more cautious approach in creating DPO data. We generate a smaller volume of DPO data, ensuring it is fully reviewed, edited and/or re-written by humans. From our initial experiments with small toy datasets, we observed visible issues even with **0.1**% of noisy seed DPO data. However, after **scaling**, there is a possibility that the model can ignore this noisy text.

There are two approaches for generating negative outputs: *(i)* on-policy: we use the generations of the model we are tuning as negative outputs, and *(ii)* off-policy: we use another, roughly similar, model to generate the negative outputs. We did not verify that the negative outputs were worse than the positive; we ensured that the positive outputs were of the highest quality, such that they were almost always better than the negative outputs.

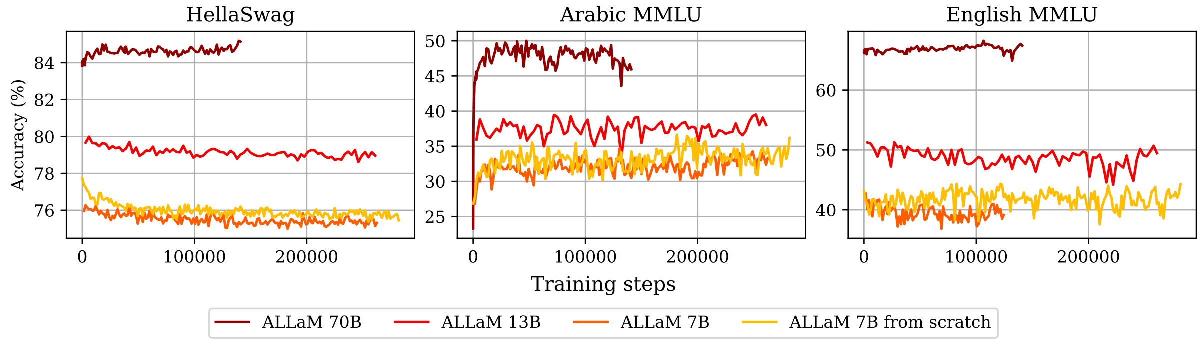


Figure 8: Benchmark evaluations throughout ALLaM model training. Using HellaSwag as a proxy for language understanding, seems that smaller models' performance reduce when introducing Arabic, while larger models (70b) have enough capacity to improve simultaneously in English and Arabic. Arabic language acquisition is rapid in all models, as indicated by Arabic MMLU.

Compared to pretraining and SFT, the model is most sensitive to the DPO data. Therefore, we ensured the highest quality data are collected and verified. In early DPO models, we did not verify all the samples, and found that moderately noisy samples resulted in broken models that repeat generations, or output incoherent text.

4 Evaluation

In this section, we dive deeper into the evaluation of our model and report the results of our validations of ALLaM 7B, 13B and 70B models, as well as relevant models such as `GPT-4`, `Command-R+` (Gomez, 2024), `Jais-30B` (Sengupta et al., 2023), and others. Our evaluation mechanisms integrate three key aspects: *(i)* automatic evaluations, *(ii)* LLM-based evaluations, *(iii)* human evaluations.

Limitations We start discussing evaluation by stating current limitations. Recently (Alzahrani et al., 2024) showed that multiple choice or cloze test based evaluation can be tricky and flip the benchmark. In addition to that MT-bench uses LLM as a judge and reportedly has high contamination possibility. Additionally, doing human evaluation is time consuming and requires training human evaluators. In this work, we try to ensure robust validation and attain a balanced assessment of the quantitative metrics and qualitative effectiveness and relevance of models in various applications and domains.

4.1 Automatic Evaluations

Figure 8 shows the *continuous evaluation* of our pretraining. Table 4 and 5 give an overview of the performance of ALLaM-instruct models compared to the relevant models. More detailed results can be found in Table 8, and 9. In Arabic benchmarks, we can see that ALLaM 70B scores are the best in five (MMLU arabic both versions, Exams, ETEC, araTruthfulQA) out of the eight benchmark sets. For the remaining benchmarks: araSwag Jais 30B v3 scored the best (for this dataset, it is not publicly available but the authors shared with us the training and dev set and we are reporting on the dev set); ACVA ALLaM 7B scored the best and for araMath LLamaa3 70B scored the best with ALLaM 70B scoring second best. In English benchmarks, we can see a high competition between ALLaM 70B and LLama 3 70B, where LLama3 70B scored the best in seven (MMLU, MMLU-Pro, Ethics, TruthfulQA, ARC, MixEval (hard - standard)) out of the nine benchmark sets and ALLaM 70B scoring second best in five of these (MMLU, MMLU-Pro, ARC, MixEval (hard-standard)). For the Ethics benchmark ALLaM 13B scored second best and for TruthfulQA Mistral 7B scored second best. As for the remaining two benchmarks AGIEval and HellaSwag ALLaM 70B scored the best.

4.2 LLM-based Evaluations

MT Bench (Zheng et al., 2024) consists of 80 multi-turn questions to evaluate models' capabilities and complex instruction-following. In addition to the English version, we created an Arabic version of MT Bench developed via human translation and localization. GPT-4 serves as the LLM judge, scoring responses as recommended in (Zheng et al., 2024). Model performance is compared turn by turn, with results shown in Table 6, where ALLaM 70B achieves the best Arabic performance.

4.3 Human Evaluation

Finally we perform human evaluations to gather voting and calculate ELO scores. We developed an Arabic multi-turn dataset that covers seven domains: Arabic linguistics, history, health, politics,

		araSwag	ACVA	MMLU (ar)		Exams (ar)	ETEC	araTruthfulQA	araMath
				Koto et al. (2024)	Huang et al. (2023)				
		10-shot	5-shot	0-shot	0-shot	5-shot	0-shot	0-shot	5-shot
ALLaM-Instruct	7B	49.28	**80.33**	66.9	49.6	52.7	62.95	36.4	36.5
AceGPT-Chat	7B	43.4	59.35	45.8	33.58	35.57	36.05	37.9	22.5
Llama 2-Chat	7B	24.44	52.46	33.33	26.45	25.33	26.69	29.9	21.5
Mistral-Instruct-v0.3	7B	30.59	60.7	44.3	34.06	31.1	34.41	30.3	26.0
Llama 3-Instruct	8B	33.99	75.21	53.98	41.49	44.32	49.42	34.0	38.3
ALLaM-Instruct	13B	54.77	78.59	68.11	51.03	54.93	65.59	37.5	46.8
Llama 2-Chat	13B	25.75	60.14	35.84	28.73	22.91	30.44	31.4	22.3
Jais-Chat	13B	77.12	70.68	54.8	41.43	46.93	48.68	31.6	25.3
ALLaM-Instruct	70B	57.91	79.01	**75.92**	**62.23**	**58.47**	**78.38**	**38.4**	56.8
Jais-Chat-v3	30B	**88.37**	70.05	62.37	30.15	51.21	38.53	37.3	32.5
Llama 2-Chat	70B	30.72	59.49	40.77	32.86	28.68	30.6	32.3	25.5
Llama 3-Instruct	70B	45.75	80.26	36.27	60.11	**58.47**	71.41	37.7	**59.70**

Table 4: Comparison of Arabic benchmarks for various instruct models.

		AGIEval	MMLU	MMLU-Pro	Ethics	TruthfulQA	ARC	HellaSwag	MixEval	
			Average				Challenge		Hard	Standard
		0-shot	0-shot	CoT 5-shot	0-shot	0-shot	0-shot	0-shot	5/0-shot (base/ft)	5/0-shot (base/ft)
ALLaM-Instruct	7B	47.09	58.31	27.78	69.8	42.11	51.45	75.2	28.9	67.6
AceGPT-Chat	7B	26.33	44.53	—	53.38	49.34	42.32	70.92	—	—
Llama 2-Chat	7B	35.55	46.4	22.87	58.88	45.32	44.28	75.52	30.8	61.7
Mistral-Instruct-v0.3	7B	42.22	59.75	36.33	73.59	59.65	58.7	82.88	36.2	70.0
Llama 3-Instruct	8B	44.35	63.82	41.32	68.07	51.72	56.83	75.81	45.6	75.0
ALLaM-Instruct	13B	48.42	61.8	34.05	76.47	57.69	55.89	81.14	37.2	72.8
Llama 2-Chat	13B	37.73	53.3	27.19	70.52	43.95	50.17	79.66	—	—
Jais-Chat	13B	31.45	49.46	—	64.92	39.66	46.84	77.6	—	—
ALLaM-Instruct	70B	**65.67**	75.43	48.61	76.16	58.78	59.56	**84.97**	51.60	83.5
Jais-Chat-v3	30B	36.78	57.57	26.45	68.03	42.34	51.02	78.91	—	—
Llama 2-Chat	70B	46.0	61.15	35.16	68.5	52.77	54.27	82.14	38.0	74.6
Llama 3-Instruct	70B	63.78	**78.38**	**59.52**	**77.09**	**61.79**	**64.33**	82.49	**55.90**	**84.00**

Table 5: Comparison of English benchmarks for various instruct models.

Model	English			Arabic		
	Avg.	Turn 1	Turn 2	Avg.	Turn 1	Turn 2
AceGPT 13B-chat	5.44	6.76	4.12	6.33	7.01	5.64
ALLaM 13B Instruct	7.34	7.67	7.01	7.57	7.9	7.23
ALLaM 70B Instruct	7.44	**7.91**	6.96	**8.19**	**8.4**	**7.97**
Jais 13B Chat	4.18	4.39	3.96	4.72	5.07	4.36
Jais 30B Chat v1	3.89	4.13	3.64	3.54	4.13	2.95
Jais 30B Chat v3	5.86	6.25	5.47	6.28	6.78	5.78
Cohere Command R+	7.41	7.63	7.18	7.97	8.28	7.65
Cohere Command R	6.99	7.19	6.79	7.47	7.82	7.12
DBRX Instruct	7.16	7.33	6.98	7.83	8.19	7.46
GPT 3.5 Turbo	**7.55**	7.79	**7.31**	8.12	8.39	7.84

Table 6: MT Bench scores for Arabic and English. The scores represent the average GPT judge score over the 80 samples ranging from 0 to 10.

coding, entertainment, and ethics, each domain contains ten questions with two turns. Each comparison was evaluated by three evaluators, and we calculated the majority voting among them. In cases of disagreement, a fourth evaluator was used to break the tie. ALLaM 13B win rate was always higher than its loss rate compared with other models. Figure 9 shows the ELO scores of the human evaluations. ELO scoring had two configuration, the default scoring rewards the good model with 1 point, the tie (good and both-bad) with 0.5 points, penalizing the bad model. The custom configuration, however, penalizes the bad model and both models if both models provided bad responses. From the figure, GPT-4 achieved the highest score, followed by ALLaM 13B with the second highest score, outperforming (or matching) larger models such as CommandR+.

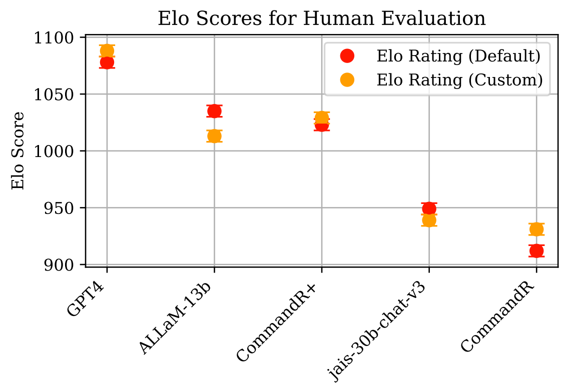


Figure 9: ELO Scores for Human Evaluation Across Various Models

5 Conclusion

ALLaM model series mark a significant leap in Arabic Language Technologies (ALT) by achieving state-of-the-art performance across various Arabic benchmarks and enhancing English performance. Through careful training that emphasizes language alignment and transferability, our models demonstrate effective second-language acquisition without catastrophic forgetting. The strategic use of translation data, knowledge encoding, and alignment with human preferences have been crucial in this success. Our openly available models on the redacted aim to support and enrich the cultural and technological landscape of the Arab World, fostering further advancements in LLMs.

6 Limitations

The model was trained on data that may potentially include toxic language, unsafe content, and societal biases originally sourced from the internet, leading to the possible amplification of these biases and toxic responses, particularly when prompted with toxic inputs. Although the model underwent concise safety training during the alignment phase, more community feedback is needed to iteratively improve the model. Additionally, inherent uncertainties in generative models mean that trials cannot encompass every possible use case, making it impossible to predict the model's responses in all contexts. This can occasionally result in inaccurate, biased, or socially unacceptable outputs, even if the prompt itself is not explicitly offensive. Developers must conduct thorough safety evaluations and make specific adjustments to ensure the model is suitable for its intended purposes. Furthermore, the output generated by this model should not be considered a statement from the model's creators or any affiliated organization.

7 Ethical Statement

While conducting and presenting this research, we are committed to upholding the highest ethical standards. We recognize the potential impact of large language models on society and the importance of ensuring their responsible development and deployment. Our work adheres to principles of fairness, transparency, and inclusivity, striving to mitigate biases and ensure diverse representation in our training data. We are mindful of privacy concerns and have taken steps to anonymize and secure data used in our research. Additionally, we acknowledge the potential for misuse of language technologies and advocate for their ethical application, promoting beneficial use cases while being vigilant about unintended consequences. Our models are made openly available to foster collaboration and further research, with the aim of contributing positively to the advancement of language technologies and supporting the cultural and technological growth of the Arabic-speaking world.

8 Risk Statement

The deployment and use of LLMs in various applications pose significant risks, including data privacy and security concerns due to the inadvertent inclusion of sensitive information in training datasets. LLMs can perpetuate or amplify biases, resulting in unfair treatment and discrimination in critical decision-making processes. They can also generate convincing but inaccurate content, spreading misinformation and potentially influencing public opinion negatively. Over-reliance on LLMs may diminish human judgment, and the models' susceptibility to adversarial attacks can compromise system integrity. To mitigate these risks, we follow robust governance, continuous monitoring, and iterative improvements. We also adhere to best practices in data handling and model training, fostering transparency and accountability in LLM development.

References

Muhammad Abdul-Mageed, Abdelrahim Elmadany, Alcides Inciarte, Md Tawkat Islam Khondaker, et al. 2023. Jasmine: Arabic gpt models for few-shot learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16721–16744.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

Asaad Alghamdi, Xinyu Duan, Wei Jiang, Zhenhai Wang, Yimeng Wu, Qingrong Xia, Zhefeng Wang, Yi Zheng, Mehdi Rezagholizadeh, Baoxing Huai, et al. 2023. Aramus: Pushing the limits of data and model scale for arabic natural language processing. *arXiv preprint arXiv:2306.06800*.

Reem Alghamdi, Zhenwen Liang, and Xiangliang Zhang. 2022. ArMATH: a dataset for solving Arabic math word problems. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 351–362, Marseille, France. European Language Resources Association.

Alibaba. 2024. Qwen2 technical report.

Norah Alzahrani, Hisham Abdullah Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yusef Almushaykeh, Faisal Mirza, Nouf Alotaibi, Nora Altwairesh, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. When benchmarks are targets: Revealing the sensitivity of large language model leaderboards. *Preprint*, arXiv:2402.01781.

Amazon. 2021. Sustainability in the cloud.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng

Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report. *Preprint*, arXiv:2305.10403.

Anthropic. 2022. The claude 3 model family: Opus, sonnet, haiku.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Aragpt2: Pre-trained transformer for arabic language generation. *arXiv preprint arXiv:2012.15520*.

M Saiful Bari, Shafiq Joty, and Prathyusha Jwalapuram. 2020. Zero-resource cross-lingual named entity recognition. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pages 7415–7423.

Loubna Ben Allal, Niklas Muennighoff, Logesh Kumar Umapathi, Ben Lipkin, and Leandro von Werra. 2022. A framework for the evaluation of code generation models. `https://github.com/bigcode-project/bigcode-evaluation-harness`.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

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*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *Preprint*, arXiv:2303.12712.

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 NAACL-HLT 2019*.

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*, abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Jesse Dodge, Taylor Prewitt, Remi Tachet Des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A. Smith, Nicole DeCario, and Will Buchanan. 2022. Measuring the carbon intensity of ai in cloud instances. *Preprint*, arXiv:2206.05229.

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. *arXiv preprint arXiv:2203.10945*.

AbdelRahim Elmadany, Muhammad Abdul-Mageed, et al. 2022. Arat5: Text-to-text transformers for arabic language generation. In *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long papers)*, pages 628–647.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2021. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027.

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. 2023. A framework for few-shot language model evaluation.

Aidan Gomez. 2024. Introducing Command R+: A Scalable LLM Built for Business.

Google. 2021. Carbon free energy for google cloud regions.

Google. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Momchil Hardalov, Todor Mihaylov, Dimitrina Zlatkova, Yoan Dinkov, Ivan Koychev, and Preslav Nakov. 2020. Exams: A multi-subject high school examinations dataset for cross-lingual and multilingual question answering. *arXiv preprint arXiv:2011.03080*.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Dan Hendrycks and Mantas Mazeika. 2022. X-risk analysis for ai research. *Preprint*, arXiv:2206.05862.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

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*.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Mohammad Abdullah Matin Khan, M Saiful Bari, Xuan Long Do, Weishi Wang, Md Rizwan Parvez, and Shafiq Joty. 2023. xcodeeval: A large scale multilingual multitask benchmark for code understanding, generation, translation and retrieval. *Preprint*, arXiv:2303.03004.

Denis Kocetkov, Raymond Li, Loubna Ben Allal, Jia Li, Chenghao Mou, Carlos Muñoz Ferrandis, Yacine Jernite, Margaret Mitchell, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, Leandro von Werra, and Harm de Vries. 2022. The stack: 3 tb of permissively licensed source code. *Preprint*, arXiv:2211.15533.

Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Boda Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, et al. 2024. Arabicmmlu: Assessing massive multitask language understanding in arabic. *arXiv preprint arXiv:2402.12840*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Imad Lakim, Ebtesam Almazrouei, Ibrahim Abualhaol, Merouane Debbah, and Julien Launay. 2022. A holistic assessment of the carbon footprint of noor, a very large arabic language model. In *Proceedings of BigScience Episode# 5–Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 84–94.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets. *Preprint*, arXiv:2305.18486.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy

Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models.

Peiqin Lin, Shaoxiong Ji, Jörg Tiedemann, André FT Martins, and Hinrich Schütze. 2024. Mala-500: Massive language adaptation of large language models. *arXiv preprint arXiv:2401.13303*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Alexandra Sasha Luccioni and Alex Hernandez-Garcia. 2023. Counting carbon: A survey of factors influencing the emissions of machine learning. *Preprint*, arXiv:2302.08476.

Gary Marcus. 2022. Is chatgpt really a "code red" for google search?

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

Mistral. 2024. Au large.

El Moatez Billah Nagoudi, Muhammad Abdul-Mageed, AbdelRahim Elmadany, Alcides Alcoba Inciarte, and Md Tawkat Islam Khondaker. 2022. Jasmine: Arabic gpt models for few-shot learning. *arXiv preprint arXiv:2212.10755*.

Tarek Naous, Michael J. Ryan, Alan Ritter, and Wei Xu. 2024. Having beer after prayer? measuring cultural bias in large language models. *Preprint*, arXiv:2305.14456.

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

Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng, Mahir Shah, Kabir Jain, Graham Neubig, and Yang You. 2024. Mixeval: Deriving wisdom of the crowd from llm benchmark mixtures. *Preprint*, arXiv:2406.06565.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

Aitor Ormazabal, Che Zheng, Cyprien de Masson d'Autume, Dani Yogatama, Deyu Fu, Donovan Ong, Eric Chen, Eugenie Lamprecht, Hai Pham, Isaac Ong, Kaloyan Aleksiev, Lei Li, Matthew Henderson, Max Bain, Mikel Artetxe, Nishant Relan, Piotr Padlewski, Qi Liu, Ren Chen, Samuel Phua, Yazheng Yang, Yi Tay, Yuqi Wang, Zhongkai Zhu, and Zhihui Xie. 2024. Reka core, flash, and edge: A series of powerful multimodal language models. *Preprint*, arXiv:2404.12387.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *Preprint*, arXiv:1802.05365.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Szymon Ruciński. 2024. Efficient language adaptive pre-training: Extending state-of-the-art large language models for polish. *arXiv preprint arXiv:2402.09759*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. *Preprint*, arXiv:1904.01557.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, et al. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Claude E Shannon. 1951. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

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. *CoRR*, abs/2402.00159.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *Preprint*, arXiv:1906.02243.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Merrill Swain and Sharon Lapkin. 1995. Problems in Output and the Cognitive Processes They Generate: A Step Towards Second Language Learning. *Applied Linguistics*, 16(3):371–391.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Preprint*, arXiv:1706.03762.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from language models. *Preprint*, arXiv:2112.04359.

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*.

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.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *Preprint*, arXiv:2304.06364.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. Lima: Less is more for alignment. *Preprint*, arXiv:2305.11206.

Xuhui Zhou, Zhe Su, Tiwalayo Eisape, Hyunwoo Kim, and Maarten Sap. 2024. Is this the real life? is this just fantasy? the misleading success of simulating social interactions with llms. *Preprint*, arXiv:2403.05020.

A Appendix

Contents

B Related Work

The most prominent Arabic-focused LLMs are:

1. Jais (Sengupta et al., 2023): 13b and 30b base and chat models trained from scratch using a combination of natural and translated Arabic data along with English and code data.

2. AceGPT (Huang et al., 2023): 7b and 13b base and chat models trained from Llama 2 *without* vocabulary expansion. They also highlight the dangers of using translated data on LLM localization.

While Jais and AceGPT are the most prominent ones, early open models such as AraGPT (Antoun et al., 2020), AraT5 (Elmadany et al., 2022), AraBART (Eddine et al., 2022), and Noon[8] models that utilized limited resources to serve Arabic and fueled the ambition to pursue Arabic focused models.

Other closed models such as Noor (Lakim et al., 2022), Jasmine (Abdul-Mageed et al., 2023), and Aramus (Alghamdi et al., 2023) are worth mentioning to show the interest in serving a language with over 400 million speakers worldwide.

Language adaptation of open models to other languages has been investigated in many research papers, some focus on languages written Latin scripts, which lessens the need for vocabulary expansion, such as Polish (Ruciński, 2024), in their work they adapted Mistral 7B. Mala-500 is another effort to expand to 534 languages, they expanded the vocabulary to 260K tokens, and further pretrained Llama 2 using LoRA adaptors (Lin et al., 2024), they used significantly less data for each language, and the evaluation of the approach was limited to measuring perplexity, and automatic classification benchmarks. (Cui et al., 2023) introduced Chinese Language adaptation of Llama and Alpaca models, where the vocabulary was increased to 50K tokens, then continued to pretrain the models and finally finetune them.

It is worth noting that low-resource no longer means low in data, more significantly it also means low in compute, our work has certainly benefited from the open-source community, and our direction is to provide ALLaM models as open-source pending final checks and approvals.

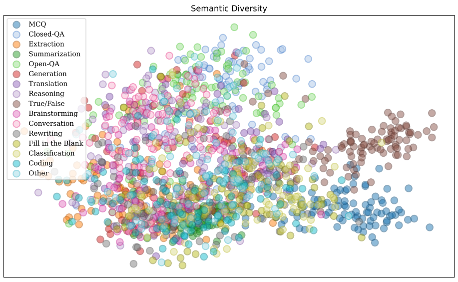


Figure 10: The semantic diversity of the prompts capabilities in Arabic v2 SFT data.

C Alignment Details

C.1 SFT Data Details.

In SFT data, we ensured that the prompts cover a sufficiently diverse embedding space, Figure 10 shows the diversity in capabilities for Arabic (v2) SFT data.

C.2 SFT Training Details.

We trained our base model, which was trained on 3.2 trillion (2T Llama + 1.2T ALLaM) tokens, for 3 epochs using `Ultra-Instinct-v2` with a learning rate of `5e-6` and a batch size of 1024. For assistant training, the model was not supposed to generate the prompt; therefore, we masked out our prompt tokens when calculating the loss. `Ultra-Instinct-v2` contains a substantial number of multi-turn conversations. To train on these multi-turn conversations, we performed `turn-augmentation`. Figure 11 visually explains the process of turn augmentation.

C.3 DPO Training Details.

For DPO, we used 512 batch size with $\mathtt{KL}_{penalty}$ and learning rate 9e-7 decayed to 5e-7 using Cosine Annealing learning rate scheduler.

Khan et al. (2023) demonstrated that model outputs can vary significantly depending on the sampling mechanism used. Building on this insight, we generate 10 additional samples for each instance by employing different temperature and nucleus sampling techniques. These additional samples are utilized to produce rejected samples, ensuring that our model provides more grounded responses and generalizes well across various sampling mechanisms. We then train the model for a single epoch using all the generated samples.

[8] https://huggingface.co/Naseej/noon-7b

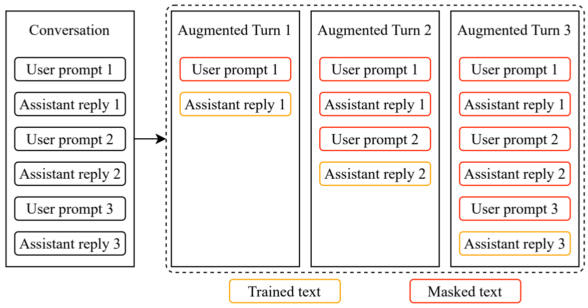


Figure 11: Augmentation process for conversations: The original conversation (left) is expanded into multiple turns (right), with user prompts and assistant replies marked for training (red) and masking (orange) to enhance the model's language understanding and multi-turn response generation capabilities.

Model	Elo rating	
	Default	Custom
GPT-4	1,078	1,088
ALLaM Instruct 13b	1,035	1,013
Command R+	1,026	1,029
Jais 30b Chat v3	949	939
Command R	912	931

Table 7: Elo rating from human evaluations on Arabic prompts.)

C.4 DPO vs PPO

One of the fundamental differences between DPO and PPO is that PPO is always on-policy with an external *Reward Model*. In our experience with DPO, we did not encounter any **significant issues** with off-policy experiments. Additionally, DPO allows for faster iteration and easier understanding of the training dynamics. The decision to use DPO over PPO was based on logistical constraints rather than a performance comparison of the algorithms. Given our compute setup and time constraints, we chose to proceed with DPO. We plan to explore PPO in future iterations of our alignment efforts.

D Evaluation Details

D.1 Human Evaluations

In the human evaluation, we have presented two models to the human for the multi-turn dataset, and we asked the human to provide their rating with a consent that their preference (without any personal identification) will be used to further improve the language model. The shared instruction with the evaluators were the following: choose model x as the winner if it had the best answer, tie if both models good, and both-bad if both models had bad responses. The model's response was considered good if if it was answering the questions correctly, have a coherent and natural language, is grammatically correct, the response is in the right language (if asked in Arabic the response should be in the same language except if the question was specifying answering in other languages.), and if the answer is aligned with human values and don't contain hate or any bias. The Elo scores for human evaluations are detailed in Table 7.

D.2 Automatic Evaluation Frameworks

Most evaluations were completed using the Language Model Evaluation Harness framework (Gao et al., 2023) with the following exceptions: HumanEval was evaluated using BigCode Evaluation Harness (Ben Allal et al., 2022). MMLU-Pro, MixEval, and Arabic MMLU (Koto et al., 2024) were evaluated using the repositories of the dataset creators.

D.3 Dataset List

The evaluation pipeline covers Arabic and English benchmarks grouped into the categories listed below:

1. Multi-domain: MixEval (Ni et al., 2024), MMLU-Pro (Wang et al., 2024), and BBH (Suzgun et al., 2022).

2. Reasoning and Commonsense: HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2019), and AraSwag (Nagoudi et al., 2022).

3. World Knowledge and Language Understanding: MMLU (Hendrycks et al., 2020),ARC Easy and Challenge (Clark et al., 2018), TriviaQA (Joshi et al., 2017), BoolQ (Clark et al., 2019), NQ Open (Kwiatkowski et al., 2019), AGIEval (Zhong et al., 2023), Exams-Ar (Hardalov et al., 2020), MMLU Arabic (tr) (Huang et al., 2023), MMLU Arabic (MBZU) (Koto et al., 2024) , and ETEC (in-house curated).

4. Safety and Alignment: Hendrycks Ethics (Hendrycks et al., 2021a), ACVA (Huang et al., 2023), TruthfulQA (Lin et al., 2022), and AraTruthfulQA (in-house curated).

5. Conversation: MT Bench (Zheng et al., 2024), and Arabic domain capability dataset (in-house curated).

6. Math: Minerva MATH (Lewkowycz et al., 2022; Hendrycks et al., 2021b), GSM8K (Cobbe et al., 2021) and araMath (in-house curated).

The following benchmarks were developed and processed in-house: ETEC is a 1891 multiple choice questions covering different exams performed by the Education and Training Evaluation Commission at KSA[9]. Additionally AraMath is a subset that focuses testing the model performance on Arabic math problems, it consists of 600 test samples that were post-processed and prepared from the AraMath dataset (Alghamdi et al., 2022). The dataset AraTruthfulQA is a dataset created using similar methodology to TruthfulQA dataset. It comprise a total of 541 samples, 285 samples were translated directly from TruthfulQA using GPT-4, then it was carefully validated and aligned to Arabic culture by human labelers. Additionally, 256 questions were curated by humans to ensure their contextual relevance and cultural appropriateness. As for MT-Bench Arabic version, we have used GPT-4 to translate the original dataset then it was reviewed and aligned to Arabic culture by human evaluators.

D.4 Detailed Results

Follow Table 8 for arabic and Table 9 for English evaluation results.

E Intended Use

ALLaM is specifically designed to expedite the research and development of ALT through Large Language Models (LLM). It serves as one of the foundational elements for building product offerings as well as facilitating experimental initiatives.

F Writing Help

We prompt ALLaM-13B-Instruct to perform grammatical check of the content.

G Computational Budget and Infra

From different stage of training we had access from 128 A100 GPUs to 1024 A100 GPUs. We trained on GPU cluster with infiniband connections to enable high-speed communication between nodes. The all-reduce test on the cluster ranges around 1200-1400 Gbps (node-node interconnect (RoCE)). The entire training period of the models are estimated around 5M GPU hours.

[9] https://etec.gov.sa/home

		araSwag	ACVA	MMLU (ar)		Exams (ar)	ETEC	araTruthfulQA	araMath
				Koto et al. (2024)	Huang et al. (2023)				
		10-shot	5-shot	0-shot	0-shot	5-shot	0-shot	0-shot	5-shot
Pretrained									
ALLaM-Base (from scratch)	7B	52.68	68.46	44.45	36.28	42.09	41.7	29.4	25.5
ALLaM-Base	7B	51.63	66.18	41.52	34.42	38.55	36.58	29.9	11.5
AceGPT	7B	46.8	59.54	36.33	27.18	32.22	25.42	30.1	19.3
Llama 2	7B	25.62	62.93	33.61	26.64	23.09	27.85	25.7	24.8
Mistral-v0.3	7B	30.33	53.81	40.81	32.1	31.47	32.45	27.0	16.3
OLMo-1.7	7B	24.44	57.8	30.97	25.7	25.7	27.17	23.5	16.8
OLMo	7B	22.09	56.07	31.41	24.98	28.31	23.1	26.2	31.7
Qwen2	7B	40.26	78.74	52.91	47.16	46.0	55.23	29.9	51.2
Gemma	7B	25.36	54.82	46.33	26.04	22.91	25.48	24.0	39.3
Llama 3	8B	38.95	71.54	47.62	38.88	44.69	42.86	29.9	43.8
ALLaM-Base	13B	54.90	77.81	51.48	40.29	47.3	44.4	28.5	17.3
Yi-1.5	9B	28.76	61.19	46.36	34.11	34.82	40.01	24.0	44.8
AceGPT-v1.5	13B	48.89	73.47	42.24	33.18	40.6	33.56	30.3	18.8
Llama 2	13B	28.63	64.52	35.83	30.0	28.86	31.13	26.2	13.8
Jais	13B	49.28	60.76	32.2	29.23	33.33	27.96	28.7	28.5
ALLaM-Base	70B	**59.35**	**79.67**	59.21	49.34	53.82	55.97	33.5	38.7
Jais-v1	30B	54.51	68.25	37.6	32.94	43.39	34.04	29.6	19.3
Jais-v3	30B	53.86	70.49	45.19	38.31	50.28	45.61	30.5	25.2
Qwen1.5	32B	37.78	73.63	55.94	48.67	49.53	57.4	34.0	45.3
Yi-1.5	34B	32.16	65.25	42.93	36.26	33.71	36.21	23.7	52.0
Mixtral-8x7B-v0.1	47B	38.43	75.64	51.25	39.74	44.32	44.61	25.5	39.8
Llama 2	70B	34.38	51.16	44.79	37.1	37.99	39.38	26.6	32.3
Llama 3	70B	54.51	74.17	36.67	59.39	55.31	64.27	31.4	53.70
Qwen1.5	72B	44.84	76.0	61.38	54.44	54.0	62.84	34.90	51.8
Qwen2	72B	51.76	68.7	**69.94**	**65**	**56.98**	**75.16**	**36**	**62.3**
DBRX	132B	47.58	72.38	53.24	47.2	47.11	51.96	26.8	49.3
Mixtral-8x22B-v0.1	141B	45.1	77.21	53.6	45.92	48.42	53.96	29.8	51.0
Fine-tuned									
ALLaM-Instruct (from scratch)	7B	50.98	79.59	69.16	51.38	52.89	67.34	30.7	42.2
ALLaM-Instruct	7B	49.28	80.33	66.9	49.6	52.7	62.95	36.4	36.5
AceGPT-Chat	7B	43.4	59.35	45.8	33.58	35.57	36.05	37.9	22.5
Llama 2-Chat	7B	24.44	52.46	33.33	26.45	25.33	26.69	29.9	21.5
Mistral-Instruct-v0.3	7B	30.59	60.7	44.3	34.06	31.1	34.41	30.3	26.0
OLMo-Instruct	7B	25.36	58.74	32.74	26.5	24.77	27.33	29.6	36.5
Qwen2-Instruct	7B	37.78	79.3	49.82	48.07	47.3	56.18	35.1	51.3
Gemma-it	7B	25.62	58.03	41.48	23.15	22.91	23.73	34.8	37.0
Llama 3-Instruct	8B	33.99	75.21	53.98	41.49	44.32	49.42	34.0	38.3
Aya-23	8B	51.11	73.65	54.37	36.39	43.76	42.28	31.6	32.0
ALLaM-Instruct	13B	54.77	78.59	68.11	51.03	54.93	65.59	37.5	46.8
Yi-1.5-Chat	9B	29.8	67.57	45.5	36.02	31.47	43.6	28.7	47.8
AceGPT-Chat-v1.5	13B	49.41	64.93	60.7	37.92	40.04	42.81	36.4	22.5
Llama 2-Chat	13B	25.75	60.14	35.84	28.73	22.91	30.44	31.4	22.3
Jais-Chat	13B	77.12	70.68	54.8	41.43	46.93	48.68	31.6	25.3
ALLaM-Instruct	70B	57.91	79.01	**75.92**	62.23	58.47	**78.38**	38.4	56.8
Jais-Chat-v1	30B	80.52	71.14	60.4	43.99	48.6	48.52	32.9	26.0
Jais-Chat-v3	30B	**88.37**	70.05	62.37	30.15	51.21	38.53	37.3	32.5
Qwen1.5-Chat	32B	37.39	78.86	57.25	50.62	48.23	59.73	39.0	43.0
Yi-1.5-Chat	34B	30.85	65.96	45.6	35.47	35.2	40.22	25.3	49.8
CommandR	35B	55.42	78.34	60.19	48.38	50.65	55.44	33.8	47.2
Aya-23	35B	55.56	79.69	57.71	47.78	51.77	56.18	33.8	43.8
Mixtral-8x7B-Instruct-v0.1	47B	37.91	77.27	52.66	41.09	42.64	49.37	32.5	39.7
Llama 2-Chat	70B	30.72	59.49	40.77	32.86	28.68	30.6	32.3	25.5
Llama 3-Instruct	70B	45.75	80.26	36.27	60.11	58.47	71.41	37.7	59.70
Qwen1.5-Chat	72B	46.8	**80.49**	64.99	54.32	53.26	62.32	42.30	45.7
Qwen2-Instruct	72B	51.9	79.98	71.51	**66.18**	**58.66**	75.16	**47.70**	**61.70**
CommandR+	104B	59.35	80.37	66.33	52.98	52.89	62.1	37.0	50.2
DBRX-instruct	132B	45.75	76.46	56.6	46.73	48.79	53.17	30.5	48.8
Mixtral-8x22B-Instruct-v0.1	141B	43.79	76.45	58.92	46.74	49.72	55.55	35.1	46.0

Table 8: Comparison of Arabic benchmarks for Various Models.

		AGIEval	MMLU					MMLU-Pro	Ethics	WinoGrande	TruthfulQA	PIQA	ARC		BoolQ	HellaSwag	TriviaQA	NQ Open	Minerva MATH	GSM8K	BBH	MixEval		HumanEval	
			STEM	Humanities	Social Sciences	Other	Average						Easy	Challenge								Hard	Standard		
		0-shot	0-shot	0-shot	0-shot	0-shot	0-shot	CoT 5-shot	0-shot	0-shot	0-shot	0-shot	0-shot	0-shot	0-shot	0-shot	0-shot	0-shot	4-shot	5-shot	CoT 3-shot	5/0-shot (base/ft)	5/0-shot (base/ft)	pass@1	pass@10
Pretrained																									
ALLaM-Base (from scratch)	7B	25.46	38.44	40.04	46.73	48.02	42.91	20.31	59.9	68.43	35.26	80.58	67.42	43.52	75.87	76.26	27.63	15.68	5.88	16.15	39.39	26.90	48.5	—	—
ALLaM-Base	7B	24.19	34.73	37.43	45.53	46.96	40.71	20.43	44.15	68.9	35.22	79.0	73.36	45.65	74.77	76.17	42.1	13.96	5.52	16.98	41.28	20.4	50.2	—	—
AceGPT	7B	26.44	30.07	35.52	37.73	38.24	35.38	—	48.09	69.3	38.39	77.75	72.64	44.11	75.66	75.43	49.64	18.39	2.84	11.14	36.81	—	—	12.28	20.23
Llama 2	7B	25.32	34.32	38.83	46.12	47.06	41.23	20.56	44.25	69.06	38.96	79.11	74.58	46.25	77.74	75.97	52.51	14.99	3.4	13.87	39.63	—	—	—	—
Mistral-v0.3	7B	32.85	49.54	52.84	68.57	66.85	58.65	31.67	64.49	73.56	42.57	82.15	78.24	52.13	82.14	80.38	60.68	20.42	12.38	37.53	58.27	—	—	27.82	42.12
OLMo-1.7	7B	30.58	39.11	43.23	54.18	52.66	46.79	18.57	60.47	69.14	35.91	80.14	75.04	45.14	80.89	77.28	15.35	11.41	5.98	27.67	34.07	—	—	17.66	27.21
OLMo	7B	22.79	28.29	27.04	28.73	28.23	27.95	—	48.25	66.38	35.85	79.43	68.77	40.36	72.48	75.65	29.69	11.61	1.9	5.23	29.66	—	—	13.05	19.26
Qwen2	7B	45.09	64.57	60.55	80.79	75.73	69.25	40.84	65.24	72.38	54.25	81.18	74.41	49.74	84.89	78.82	—	—	—	77.94	58.41	—	—	—	—
Gemma	7B	38.6	53.89	54.67	71.14	68.49	61.16	34.24	66.09	73.01	45.49	81.23	80.81	53.24	82.84	80.72	—	—	—	0.15	—	—	—	32.48	52.2
Llama 3	8B	33.87	52.93	54.62	72.9	71.13	61.9	35.54	56.58	73.48	44.03	80.9	77.78	52.99	81.13	79.14	63.35	16.4	15.86	50.72	62.91	—	—	27.5	44.61
ALLaM-Base	13B	29.14	41.42	45.78	57.07	56.49	49.64	25.78	62.63	73.32	36.36	80.79	75.67	50.77	80.09	80.13	35.73	19.36	8.02	29.49	47.89	26.2	57.80	—	—
Yi-1.5	9B	37.8	63.88	60.91	79.1	73.61	68.37	40.58	63.35	73.01	46.67	80.79	79.12	54.69	85.81	77.78	54.32	17.87	29.96	64.82	70.97	—	—	**39.06**	**56.53**
AceGPT-v1.5	13B	32.4	42.06	48.33	59.64	58.48	51.65	26.01	59.25	72.38	38.07	80.2	76.26	50.77	80.7	80.1	57.43	15.6	6.5	23.96	46.32	—	—	17.44	27.46
Llama 2	13B	32.47	40.95	47.08	58.56	57.39	50.5	25.86	60.93	72.14	36.89	80.52	77.44	48.98	80.58	79.39	60.89	14.93	5.54	23.43	47.35	—	—	17.67	28.97
Jais	13B	24.12	29.27	33.39	34.32	35.79	33.2	—	61.77	68.51	35.39	77.91	70.08	41.81	67.74	71.76	33.45	9.61	4.18	11.45	36.86	—	—	—	—
ALLaM-Base	70B	42.99	57.09	60.36	77.97	72.74	66.22	39.84	57.79	77.43	43.99	83.19	78.83	55.46	83.3	85.22	51.91	26.15	21.04	63.31	70.76	**46.20**	**77.90**	22.23	39.72
Jais-v1	30B	25.81	36.22	39.94	46.34	44.29	41.47	—	64.45	68.35	36.49	79.54	73.44	45.9	74.31	75.11	42.57	12.08	5.88	16.91	42.5	—	—	—	—
Jais-v3	30B	28.79	42.78	49.82	57.1	58.45	51.74	25.93	61.52	70.64	40.33	80.52	75.59	48.55	80.89	79.09	50.67	16.29	7.86	25.47	52.05	—	—	—	—
Qwen1.5	32B	47.02	66.67	64.06	81.87	77.95	71.62	44.7	67.37	74.27	57.46	82.75	71.38	51.02	87.31	83.7	—	—	37.56	74.91	53.6	—	—	0.8	3.0
Yi-1.5	34B	49.30	69.71	68.25	83.59	79.56	74.44	49.57	71.63	79.08	53.88	82.54	79.38	56.74	83.79	82.98	66.18	24.35	35.32	77.03	76.16	—	—	38.09	53.72
Mixtral-8x7B-v0.1	47B	39.62	58.99	60.62	78.26	74.22	67.13	41.76	63.21	75.61	48.5	83.57	83.38	59.81	85.26	83.88	71.39	29.75	27.18	58.53	69.36	—	—	37.79	55.15
Llama 2	70B	44.0	54.65	60.13	76.89	72.1	65.22	38.48	65.99	77.98	44.82	82.75	81.02	57.25	83.67	83.81	72.21	25.43	13.96	53.75	67.04	—	—	30.28	46.67
Llama 3	70B	48.04	69.71	67.95	86.68	81.56	75.46	53.88	72.61	**80.51**	45.66	84.6	85.82	64.25	85.26	84.95	77.84	23.66	37.94	81.12	**82.12**	—	—	4.63	11.74
Qwen1.5	72B	46.67	69.11	66.61	84.53	81.56	74.41	46.59	72.47	77.51	**59.63**	82.64	74.79	53.67	88.69	84.93	—	—	36.9	79.3	47.24	—	—	—	—
Qwen2	72B	**57.05**	**79.58**	**77.62**	**89.7**	**86.13**	**82.59**	**57.55**	**76.97**	79.16	54.73	83.24	80.64	60.15	89.45	85.57	—	—	**49.08**	**88.78**	79.67	—	—	—	—
DBRX	132B	44.39	63.02	63.06	80.99	78.08	70.3	44.65	65.69	76.64	55.63	**85.42**	**86.32**	**67.06**	**90.15**	**88.30**	**80.85**	**40.19**	30.46	71.42	67.52	—	—	—	—
Mixtral-8x22B-v0.1	141B	46.84	66.67	68.71	83.98	79.21	73.92	51.11	66.71	79.87	52.2	84.93	84.18	63.14	87.89	86.25	77.32	29.7	38.88	75.97	80.79	—	—	—	—
Fine-tuned																									
ALLaM-Instruct (from scratch)	7B	46.35	54.27	55.77	71.95	69.36	61.99	30.34	69.35	69.93	41.31	80.3	75.04	52.05	82.94	75.4	—	—	13.02	53.6	—	33.8	70.7	—	—
ALLaM-Instruct	7B	47.09	49.95	52.88	67.21	66.21	58.31	27.78	69.8	70.56	42.11	79.82	76.22	51.45	78.93	75.2	—	—	12.84	49.28	—	28.9	67.6	22.24	29.85
AceGPT-Chat	7B	26.33	34.89	42.47	49.89	52.14	44.53	—	53.38	66.38	49.34	75.24	68.01	42.32	75.84	70.92	—	—	3.0	13.04	35.99	—	—	17.65	27.08
Llama 2-Chat	7B	35.55	36.41	43.19	53.01	54.84	46.4	22.87	58.88	66.38	45.32	77.26	69.74	44.28	79.76	75.52	—	—	4.86	23.35	40.15	30.8	61.7	—	—
Mistral-Instruct-v0.3	7B	42.22	50.68	54.56	69.52	67.14	59.75	36.33	73.59	73.8	59.65	82.59	82.62	58.7	85.81	82.88	—	—	13.28	48.67	56.78	36.2	70.0	36.59	52.55
OLMo-Instruct	7B	35.2	38.34	42.7	55.96	54.33	47.2	—	63.54	66.77	45.55	76.77	64.1	43.43	78.62	78.54	—	—	1.5	11.52	33.31	26.7	55.0	16.6	23.54
Qwen2-Instruct	7B	53.72	63.91	62.93	80.37	75.64	69.78	46.79	71.62	69.69	57.31	80.63	76.64	54.1	85.41	80.6	—	—	—	77.86	59.39	—	—	—	—
Gemma-it	7B	39.27	42.69	44.78	58.34	57.74	50.15	25.78	68.7	69.22	47.49	77.97	72.64	48.89	81.35	73.04	—	—	—	2.27	—	39.1	69.6	—	—
Llama 3-Instruct	8B	44.35	54.71	57.96	74.13	71.71	63.82	41.32	68.07	71.67	51.72	78.67	79.76	56.83	83.09	75.81	—	—	27.02	75.44	67.67	45.6	75.0	55.20	71.92
Aya-23	8B	35.1	39.8	47.91	57.75	57.23	50.31	24.79	50.99	63.06	45.38	78.94	68.73	45.22	83.79	73.98	—	—	7.38	42.15	48.07	—	—	11.12	18.21
ALLaM-Instruct	13B	48.42	53.03	55.01	74.1	68.81	61.8	34.05	76.47	74.11	57.69	81.34	77.36	55.89	84.77	81.14	—	—	23.22	71.34	—	37.2	72.8	31.71	49.16
Yi-1.5-Chat	9B	55.93	66.51	62.66	79.4	72.9	69.46	47.29	71.46	74.43	52.49	80.58	77.06	55.46	86.61	79.22	—	—	38.28	78.85	68.61	40.9	74.2	55.1	67.62
AceGPT-Chat-v1.5	13B	38.92	46.18	53.77	68.54	66.5	58.12	31.3	71.23	71.98	48.66	80.52	75.29	52.13	82.87	80.51	—	—	7.2	31.08	49.09	—	—	21.77	35.22
Llama 2-Chat	13B	37.73	43.07	49.78	61.94	60.44	53.3	27.19	70.52	70.96	43.95	79.11	73.7	50.17	81.65	79.66	—	—	5.48	35.56	47.44	—	—	15.54	20.02
Jais-Chat	13B	31.45	39.42	47.48	55.28	56.87	49.46	—	64.92	68.51	39.66	78.24	69.4	46.84	75.81	77.6	—	—	6.46	24.64	38.1	—	—	—	—
ALLaM-Instruct	70B	65.67	68.38	71.2	84.82	79.69	75.43	48.61	76.16	79.16	58.78	84.11	79.84	59.56	87.28	84.97	—	—	30.32	81.58	70.42	51.60	83.5	40.29	58.18
Jais-Chat-v1	30B	33.1	43.32	50.58	60.16	60.22	53.18	—	69.07	69.69	42.8	80.09	79.34	52.65	87.8	79.89	—	—	9.12	32.68	44.71	—	—	—	—
Jais-Chat-v3	30B	36.78	47.99	53.37	66.4	64.92	57.57	26.45	68.03	70.09	42.34	73.18	76.3	51.02	90.0	78.91	—	—	13.68	51.25	50.79	—	—	—	—
Qwen1.5-Chat	32B	48.77	66.67	69.56	81.9	77.92	73.47	47.5	67.82	73.16	66.96	77.09	66.71	50.68	88.72	84.71	—	—	27.64	72.86	54.69	43.3	81.0	—	—
Yi-1.5-Chat	34B	61.99	69.58	70.1	83.49	79.95	75.1	53.35	77.19	79.32	62.16	80.63	80.64	61.77	90.18	84.04	—	—	41.08	81.5	75.07	51.2	81.7	**65.39**	**79.64**
CommandR	35B	46.77	52.78	58.68	73.74	70.68	63.31	38.96	69.3	70.24	56.35	81.23	77.99	57.25	86.45	82.34	—	—	11.36	59.29	62.54	45.2	77.0	—	—
Aya-23	35B	38.99	51.19	57.6	71.6	69.75	61.91	34.87	61.74	67.96	51.81	82.21	77.53	55.97	87.03	82.31	—	—	13.32	64.9	63.23	—	—	15.04	33.66
Mixtral-8x7B-Instruct-v0.1	47B	48.49	59.85	61.17	79.72	75.7	68.15	44.98	78.54	77.43	64.8	84.82	85.23	66.04	88.44	85.95	—	—	26.74	64.06	68.12	42.5	76.4	50.06	65.19
Llama 2-Chat	70B	46.0	50.68	56.28	72.51	67.88	61.15	35.16	68.5	74.9	52.77	80.96	76.26	54.27	86.73	82.14	—	—	10.9	53.15	63.05	38.0	74.6	29.59	38.93
Llama 3-Instruct	70B	63.78	72.53	72.82	87.68	83.52	78.38	59.52	77.03	75.53	61.79	82.05	84.81	64.33	87.49	82.49	—	—	**44.12**	**91.05**	**83.15**	**55.90**	84.00	—	—
Qwen1.5-Chat	72B	52.17	70.47	72.99	84.66	81.46	76.86	52.61	74.56	77.11	63.86	78.84	68.43	49.57	88.41	86.19	—	—	21.88	81.12	52.19	48.3	**84.10**	—	—
Qwen2-Instruct	72B	**66.09**	**79.42**	**77.94**	**88.89**	**85.26**	**82.29**	**62.48**	78.54	78.14	67.02	84.0	82.2	61.52	89.2	86.65	—	—	41.2	89.54	46.51	—	—	—	—
CommandR+	104B	45.97	60.23	65.33	80.31	75.41	69.7	45.22	71.58	65.82	59.64	71.27	61.78	46.33	85.81	79.73	—	—	25.34	76.04	72.91	51.4	81.5	—	—
DBRX-instruct	132B	49.09	62.99	67.93	82.26	78.89	72.38	47.75	70.9	78.37	67.23	**85.69**	**87.92**	67.75	**90.24**	**87.88**	—	—	29.88	74.15	72.66	—	—	—	—
Mixtral-8x22B-Instruct-v0.1	141B	55.5	68.25	70.75	84.4	79.88	75.2	55.92	**80.70**	**80.66**	**68.10**	85.26	85.86	**68.17**	89.88	87.70	—	—	41.80	85.6	82.86	—	—	—	—

Table 9: Comparison of English benchmarks for Various Models.

H Training Framework

At the start of the project we forked Megatron-LM and did our own customizations. During training, we utilized data, tensor, and pipeline parallelism to efficiently manage the large-scale model computations. By leveraging these parallelism techniques, we achieved significant improvements in training speed and model scalability. Our modifications also included improving data iterators, adding metadata in the checkpoints, custom data pipelines etc. Depending on how many GPUS, nodes, batchsize, overlapping strategy and parallelism, our TFlops varies in between 120 to 167. We trained our model on `bf16` mixed-precision.