Bilingual Adaptation of Monolingual Foundation Models

Gurpreet Gosal [1] **Yishi Xu** [1] **Gokul Ramakrishnan** [1] **Rituraj Joshi** [1] **Avraham Sheinin** [1]
Zhiming (Charles) Chen [1] **Biswajit Mishra** [1] **Natalia Vassilieva** [1] **Joel Hestness** [1] **Neha Sengupta** [2]
Sunil Kumar Sahu [2]

Abstract

We present an efficient method for adapting a monolingual Large Language Model (LLM) to another language, addressing challenges of catastrophic forgetting and tokenizer limitations. We focus this study on adapting Llama 2 to Arabic. Our two-stage approach begins with expanding the vocabulary and training only the embeddings matrix, followed by full model continual pre-training on a bilingual corpus. By continually pre-training on a mix of Arabic and English corpora, the model retains its proficiency in English while acquiring capabilities in Arabic. Our approach results in significant improvements in Arabic and slight enhancements in English, demonstrating cost-effective cross-lingual transfer. We also perform extensive ablations on embedding initialization techniques, data mix ratios, and learning rates and release a detailed training recipe. To establish generalizability of this language adaptation recipe we also adapted Llama 3 8B to Arabic and Llama 2 13B to Hindi.

1. Introduction

There has been a rapid advancement in open source English-dominant foundation language models like Llama 2 (Touvron et al., 2023), Mistral 7B (Jiang et al., 2023), and Llama 3, primarily trained on extensive English corpora with minimal inclusion of non-English languages. To create models proficient in low-resource languages, two approaches can be taken: training a bilingual or multilingual model from scratch or adapting an existing strong English-dominant model to the target language. While bilingual and monolingual models trained from scratch, like Jais (Sengupta et al., 2023b) and Bloom (et. al., 2023), have shown promise in non-English capabilities, they are expensive to train and have inferior capabilities in English. Adapting strong English-dominant models to new languages also pose challenges, such as catastrophic forgetting of English capabilities, inefficiencies of English-dominant tokenizers, and the need for hyperparameter adjustments. (Fujii et al., 2024; Luo et al., 2024; French, 1999; Huang et al., 2024). In this paper we focus on addressing the challenges of the model adaptation approach.

It has been shown that capabilities of Large Language Models (LLMs) such as knowledge, reasoning and truthfulness are transferable across languages (Yang et al., 2024; Sengupta et al., 2023b). This gives us the basis and motivation to explore efficient methods for cross-lingual transfer from English to Arabic through continual pre-training of a monolingual English-dominant LLM without degradation of English capabilities. Several recent works demonstrate cross-lingual transfer of foundation models (de Vries & Nissim, 2021; Marchisio et al., 2023; Csaki et al., 2023; Zhao et al., 2024; Huang et al., 2024; Da Dalt et al., 2024), yet they lack comprehensive analysis of hyperparameter tuning, tokenizer, data mix selections, and the impact of different model sizes.

We study the following aspects of cross-lingual adaptation.

Vocabulary extension We establish that adapting an existing model to a new language requires expanding the vocabulary, along with employing the methods below to maintain the model's original capabilities while acquiring new linguistic skills. We determine the optimal extension ratio of the original vocabulary through experimentation.

Embedding alignment We find that it ensuring alignment between the embeddings of the original and newly added vocabulary tokens is vital. We explore three techniques for initializing newly added token embeddings. We follow with embedding-only pre-training, which further aligns the embedding scale and orientation for original and new tokens.

Continual pre-training Following the embedding-only pre-training, we unfreeze the transformer backbone to continually pre-train the full model. We conduct experiments at the 7B model scale to assess various English-Arabic mix ratios and learning rates. We leverage the insights obtained from these experiments to perform cross-lingual adaptation to Arabic with Llama 2 13B and Llama 2 70B models.

Careful experimental study of optimal strategies for vocabulary extension, embedding alignment, Arabic and English data mixes and hyperparameter tuning, results in a recipe for

language adaptation with significant performance improvements in Arabic and, uniquely, enhancements in English on Llama 2 models. To demonstrate the generalizability of this recipe to other languages and models, we adapt Llama 3 8B to Arabic and Llama 2 13B to Hindi, which too led to significant improvements compared to the baseline.

2. Pre-training Datasets

We use the AraV5 Arabic dataset, which includes documents from various sources such as web pages, Wikipedia, Arabic news outlets, books and social media. It also includes high-quality English corpora, Books and Wikipedia, translated to Arabic. Curated by (Sengupta et al., 2023b), it was used for pre-training the Jais series of Arabic foundation models. Prior domain adaptation studies have emphasized the importance of using "replay" data, which aligns with the pre-training data domain, to preserve the foundational model's knowledge and reasoning capabilities, as proposed in works by (Gupta et al., 2023; Chen et al., 2023; Azerbayev et al., 2024). We use the Pile corpus, comprising data from 22 diverse sources including ArXiv, Wikipedia, PubmedCentral, CommonCrawl, OpenWebText, and Github (Gao et al., 2020). For Llama 2 7B adaptation, we utilize 20B tokens from AraV5, whereas for Llama 2-13B and 70B, we utilize the entire AraV5 corpus, totaling 140B tokens. For Hindi adaptation of Llama 2 13B we use a Hindi corpus of 65B tokens curated from diverse resources and mix it with Pile.

3. Methodology

3.1. Vocabulary Extension

The first step in adapting a monolingual foundation model for multilingual use is to construct a balanced vocabulary that includes all target languages. Recent state-of-the-art models such as Llama 2(Touvron et al., 2023) use byte pair encoding (BPE) (Sennrich et al., 2016) tokenizers, primarily trained on English data. These tokenizers often split non-English words into characters or bytes, creating a significant imbalance among languages. Fertility, which measures the average number of subwords produced by a single word upon tokenization, can be used to quantify this imbalance.

This imbalance introduces inefficiency in pre-training, fine-tuning and inference. Table 1 shows that the Llama 2 tokenizer needs as many as 4 times the number of tokens to represent the same Arabic text as Jais' Arabic-English bilingual tokenizer (MLV2) (Sengupta et al., 2023b). Balanced multilingual tokenizer with low fertility in all languages offers three main advantages (Petrov et al., 2023): 1) lower training and inference cost; 2) reduced latency during inference; 3) longer context windows. Models trained with low fertility tokenizers tend to perform well on downstream tasks as shown in (Ahuja et al., 2023)

We experiment with two methods, vocabulary replacement and vocabulary extension, to create balanced tokenizers for English and Arabic. Vocabulary replacement implies maintaining the base vocabulary and replacing its least frequent tokens with the most frequent Arabic tokens. Vocabulary extension adds the most frequent Arabic tokens, increasing the vocabulary size. In both methods, we ensure that the newly introduced tokens are not present in the original vocabulary. For both methods, we determine the optimal number of new tokens to create a balanced multilingual vocabulary. Using Arabic tokens from the MLV2 vocabulary, we create two candidate tokenizers and perform intrinsic and extrinsic evaluations following (Ali et al., 2024).

For intrinsic evaluation, we use fertility score to measure the efficiency of the tokenization process. We define fertility as $f = \frac{S}{W}$, where S is the total number of tokens in the tokenized text and W is the number of words in the raw text. Subsets of validation sets of Pile and AraV5 are used to calculate the English and Arabic fertility, respectively. Table 1 shows the intrinsic evaluations of two tokenizers, i) ***Llama 2-replace30***, and ii) ***Llama 2-extend100***. *Llama 2-replace30* replaces 30% of the base Llama 2 tokens while *Llama 2-extend100* extends the Llama 2 vocabulary by 100%. *Llama 2-extend100* reduces the fertility of Llama 2's tokenizer by 72.17% while maintaining the fertility in English. It also reaches a fertility in Arabic comparable to MLV2.

We perform extrinsic evaluation by continually training Llama 2 7B on a mixture of AraV5 and Pile, and monitoring the AraV5 validation loss. For a fair comparison of the tokenizers, we fix the raw text bytes at 67 GB for Pile and 345 GB for AraV5. Using *Llama 2-extend100* as the candidate tokenizer and *Llama 2* as the baseline, we tokenize the raw corpora and continually pre-train Llama 2 7B. Although the base Llama 2 tokenizer achieves a lower AraV5 validation loss compared to *Llama 2-extend100* (see Table 1), it is trained on significantly more Arabic tokens due to its ≈ 3.5 times higher fertility in Arabic. In an iso-token comparison, where the number of AraV5 tokens is fixed, *Llama 2-extend100* outperforms base Llama 2 tokenizer by $\approx 2\%$. Considering both the intrinsic and extrinsic evaluations, we select *Llama 2-extend100*. We correct all losses to align with the Llama 2 tokenizer (see A and B). We performed a similar analysis for Llama 3 8B adaptation to Arabic. Llama 3 8B has a four times larger vocabulary than Llama 2 and possesses superior multilingual capabilities including in Arabic. Consequently, Llama 3 tokenizer vocabulary has a relatively large number of Arabic tokens compared to Llama2, and we found that extending it by 20% brought the Arabic fertility within 10% of the MLV2 tokenizer's fertility rate for Arabic. For Hindi adaptation of Llama 2 we extended the base vocabulary by 100% just as

for Llama 2 Arabic adaptation.

3.2. Embedding initialization

For *Llama-extend100*, we add 32000 new Arabic tokens to the Llama 2 vocabulary, expanding the embedding and unembedding layers as $[32000, d] \rightarrow [64000, d]$ where d is the hidden size of the transformer decoder block. Our studies showed that the choice of embeddings initialization for newly added tokens is critical such that tokens which represent similar concepts in two languages are closer in the embeddings latent space. Simple methods such as Xavier (Glorot & Bengio, 2010) or Kaiming (He et al., 2015) initialization, or using the mean of the embedding vectors of Llama 2 tokens, do not yield satisfactory results. Therefore, we explore alternative methods, described below, which demonstrated superior performances.

Similarity-Based Token Embedding Initialization
This method is inspired by the approach proposed in (Minixhofer et al., 2022). For each new token, we identify the top-k similar tokens using cosine similarity in the base vocabulary, using an external embedding. We use OpenAI's `text-embedding-3-large` embeddings (Kusupati et al., 2024) for their superior quality and multilingual performance. Using cosine similarity, we find top-k similar tokens for the new token and initialize the new token embeddings by taking the weighted average of base embeddings of these similar tokens. After experimenting with different values for the k, we achieve the best results with $k = 5$. This initialization method was used for embeddings and unembeddings layers of Llama 2 13B adapted to Hindi.

Embedding Space Transformation
In this initialization method, we leverage the pre-trained embedding vectors of Jais-30B (Sengupta et al., 2023a). We use 21377 embedding vectors corresponding to tokens present in the intersection of the Llama 2 and Jais vocabularies to transform the Jais embeddings of the added tokens to the Llama 2 embedding space. Let $\mathbf{E_{Jais}}$ and $\mathbf{E_{Llama2}}$ to denote the embedding matrices of the overlapping tokens of Jais and Llama 2, $\mathbf{E_{Jais}} \in R^{21377 \times 7168}$ and $\mathbf{E_{Llama2}} \in R^{21377 \times 4096}$. We find a linear transformation to project $\mathbf{E_{Jais}}$ to $\mathbf{E_{Llama2}}$'s space by solving for W and b using the least squares method, $W\mathbf{E_{Jais}} + b = \mathbf{E_{Llama2}}$.

We find W and b such that the Euclidean ℓ_2 norm $\|W\mathbf{E_{Jais}} + b - \mathbf{E_{Llama2}}\|_2$ is minimized. The parameters W and b are then used to project added tokens into the Llama 2 embedding space. This method performs better than similarity-based initialization (see C).

3.3. Embedding-only pre-training

Even with Embedding Space Transformation initalization, the scale and the orientation of English and resulting Arabic embeddings are not aligned. Following (de Vries & Nissim, 2021), we do embedding-only pre-training using 15 billion tokens of AraV5 and Pile, mixed in 9:1 ratio. During this stage, gradient updates are applied to the embedding and the unembedding layers only, while keeping the other layers frozen. In our experiments, this method resulted in up to 2% improvement in upstream loss for Arabic.

3.4. Hyper-parameter tuning

Hyperparameter sweep is important to determine best hyperparameters such as learning rate (lr), warm-up schedule and batch size (bs). We use a linear warm-up for 1% of the total steps followed by cosine decay (Loshchilov & Hutter, 2017) to 1/10th of the peak learning rate. We compare batch sizes of 4M tokens and 6M tokens but don't see a significant difference in upstream losses. We pick 4M tokens as the final batch size. Following (Gupta et al., 2023), we setup a learning rate sweep taking three different learning rates in different ranges. Let lr_{peak} be the peak Llama 2 learning rate which is $3e$-4, we "re-warm" the learning rate to i) lr_{peak}, ii) $lr_{peak}/2$, and iii) $lr_{peak}/4$. These experiments use 14 billion AraV5 tokens mixed with Pile in $1:1$, $3:1$ and $9:1$ ratios. Across all ratios we find that lr_{peak} performs the best as shown in table 2. We used same hyperparameters for Hindi adaptation of Llama 2 13B. For Llama 3 8B we performed a similar sweep and found that for stable adaptation to new domain. a smaller learning rate of $7.5e$-5, a larger batch size of 6M tokens is required.

3.5. Data mixture

Domain adaptation involves continual pre-training a foundation model on new data not seen during the pre-training. When this new domain data is out-of-distribution, it can cause significant forgetting of prior capabilities which is referred to as *stability gap* (Guo et al., 2024). Adding a small proportion of replay data, which is closer in distribution to the pre-training data, can mitigate the forgetting. We conduct exhaustive experiments to find a minimum proportion of Pile data that should be mixed with AraV5 to mitigate forgetting. Table 2 shows results from the experiments with different data mixes. We found that mixing 1 part English with 9 parts Arabic ($1:9$ En:Ar) is sufficient to mitigate forgetting. We also don't see any forgetting in downstream evaluation as discussed in section 4. Interestingly, increasing the amount of English data while keeping Arabic tokens constant improves Arabic performance, indicating cross-lingual capability transfer. We leave the exploration of cross-lingual transfer to future work.

For Arabic adaptation of Llama 3 8B we found that a higher proportion of replay data is necessary to lower the stability gap. Therefore, $1:1$ English to Arabic dataset mixture was used where we saw cross-lingual capability transfer between

	Llama 2	MLV2 (Jais)	Llama 2-replace30	Llama 2-extend100
vocab size	32,000	84,992	32,000	64,000
En. Fertility	1.86	1.62 (-12.63%)	1.92 (+3.73%)	**1.85 (-0.03%)**
Ar. Fertility	5.06	1.29 (-74.55%)	1.66 (-67.26%)	**1.41 (-72.17%)**
Ar val. loss	**0.6371**	-	0.6440	0.6539
IsoToken Ar val. loss	0.6668	-	-	**0.6539**

Table 1. Tokenizer intrinsic and extrinsic evaluation. We see that MLV2 tokenizer reduces the fertility in Arabic by 75.55%, *Llama 2-replace30* by 62.26%, *Llama 2-extend100* by 72.17% compared to Llama 2 tokenizer.

lr	Mix En:Ar	Total tokens	AraV5 Tokens	Pile Tokens	Pile eval loss	AraV5 eval loss
7.5e-5	1:9	15.6B	14B	1.56B	1.5397	2.5965
1.5e-4	1:9	15.6B	14B	1.56B	1.5396	2.3741
3e-4	1:9	15.6B	14B	1.56B	1.546	**2.28**
7.5e-5	1:3	18.67B	14B	4.67B	1.5225	2.556
1.5e-4	1:3	18.67B	14B	4.67B	1.5205	2.3628
3e-4	1:3	18.67B	14B	4.67B	1.5234	**2.24**
7.5e-5	1:1	28B	14B	14B	1.5044	2.5135
1.5e-4	1:1	28B	14B	14B	1.5007	2.349
3e-4	1:1	28B	14B	14B	1.5093	**2.2135**

Table 2. Llama 2-7B learning rate ablation experiments with different English to Arabic data mixtures. Data was tokenized with *Llama 2-extend100* tokenizer and embeddings were initialized with subword-mean 3.2 approach. Base Llama 2-7B's Pile and AraV5 validation loss is 1.5466 and 2.95, respectively.

the two languages similar to Llama 2 adaptation. We also found that a specialized mix of English corpora constituting synthetic textbook dataset, math, coding and reasoning is required for the replay data to mitigate forgetting. For Hindi adaptation of Llama 2 13B we used $1:9$ English to Hindi mix.

4. Results

Using the methodology described in section 3 we adapt Llama 2 7B, 13B and 70B models to Arabic. We use linearly warm up of the learning rate to lr_{peak} for the first 1% of the tokens followed by cosine decay to $1/10$th of the lr_{peak}. For 7B and 13B models, we use $1:1$ En:Ar mix as we show in section 3.5 that higher proportion of English also improves Arabic performance. For Llama 2 70B we use $1:9$ En:Ar mix to reduce compute time. Llama 2 7B adaptation uses 20 billion tokens each from AraV5 and Pile. Llama 2-13B and Llama 2 70B use all 140 billion tokens from AraV5 and 140 billion and 15.56 billion tokens from Pile, respectively. Tables 3 and 4 show the 0-shot downstream performance of the resulting models against the base models and other Arabic models like Jais (Sengupta et al., 2023a) and AceGPT (Huang et al., 2024). For Arabic evaluations we translated the English downstream tasks datasets using a similar approach as in (Sengupta et al., 2023b).

We evaluated the models on the **World Knowledge tasks** MMLU (Hendrycks et al., 2021) RACE (Lai et al., 2017) and Exams (Hardalov et al., 2020); **Commonsense reasoning tasks** Hellaswag (Zellers et al., 2019), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), BoolQ (Clark et al., 2019), Arc Challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018) and Winogrande (Sakaguchi et al., 2019); **Misinformation and Bias tasks** TruthfulQA (Lin et al., 2021) and CrowS-Pairs(Nangia et al., 2020).

For all models we see improvement in Arabic tasks - 7.5% improvement in Arabic MMLU for *Llama2 70B adapted* compared to *Llama2 70B* (see 3), while the smaller models (7B and 13B) demonstrate 2% improvement in MMLU Arabic. This can be attributed to the lower token-per-parameter training regime resulting in less degradation from over-training (Hoffmann et al., 2022; Dey et al., 2023). We also observe slight improvement in average scores in English for *Llama2 70B adapted* (see4). Furthermore, compared to the state-of-the-art Arabic models, namely Jais and AceGPT, Arabic adapted Llama2 models significantly improve on Arabic downstream tasks.

For Arabic adaptation Llama 3 8B was continually pre-trained for 65B tokens with $1:1$ mix of English and Arabic using hyper-parameters as discussed in 3.4. In 5 and 6 we depict the performance of *Llama 3 8B adapted* on Arabic and English tasks, respectively, and a comparison against the Llama 3 8B base model. We see 4.5% improvement on average in Arabic downstream tasks compared to the base model performance while a 1% improvement in downstream English tasks.

Llama 2 13B was trained on 72B tokens with $1:9$ English to Hindi dataset mixture. Learning rate was re-warmed to the pre-training peak learning rate of $3e$-4 and a batch size of 4M tokens. We evaluate on XNLI-Hindi (Conneau et al.,

Model	Knowledge Average	Commonsense Reasoning Average	Misinformation, Bias Average	Overall Average
llama 2-70b adapted	38.4%	52.1%	51.4%	49.2%
llama 2-13b adapted	34.1%	49.2%	48.8%	46.1%
llama 2-7b adapted	33.8%	46.1%	49.1%	43.5%
llama 2-70b base	31.4%	42.8%	48.9%	41.8%
llama 2-13b base	29.4%	40.3%	47.7%	39.6%
llama 2-7b base	27.3%	39.3%	47.5%	38.5%
Acegpt-13b	32.3%	45.4%	50.8%	43.8%
Jais 30b	38.0%	52.1%	51.2%	49.1%

Table 3. Arabic summarized comparisons between: (1) Llama 2 models pre and post adaptation; (2) other notable English-Arabic bilingual models: Jais 30b (Sengupta et al., 2023a) and AceGPT (Huang et al., 2024)

Model	Knowledge Average	Commonsense Reasoning Average	Misinformation, Bias Average	Overall Average
llama 2-70b adapted	45.9%	66.6%	56.0%	60.9%
llama 2-13b adapted	38.4%	62.8%	49.3%	55.9%
llama 2-7b adapted	36.2%	57.8%	51.6%	52.3%
llama 2-70b base	48.8%	64.4%	57.1%	60.2%
llama 2-13b base	37.9%	60.8%	53.7%	55.3%
llama 2-7b base	36.1%	58.9%	55.4%	54.1%
Acegpt-13b	37.2%	62.0%	56.6%	56.5%
Jais 30b	41.3%	64.6%	56.3%	58.8%

Table 4. English summarized comparisons between: (1) Llama 2 models pre and post adaptation; (2) other notable English-Arabic bilingual models: Jais 30b (Sengupta et al., 2023a) and AceGPT (Huang et al., 2024)

Model	Knowledge Average	Commonsense Reasoning Average	Misinformation, Bias Average	Overall Average
llama 3-8b base	33.0%	45.2%	47.3%	43.2%
llama 3-8b adapted	38.8%	49.2%	53.3%	47.9%

Table 5. Arabic summarized comparisons between Llama 3 8B model pre and post adaptation where we see a 4.5% average improvement across all domains

Model	Knowledge Average	Commonsense Reasoning Average	Misinformation, Bias Average	Overall Average
llama 3-8b base	51.1%	63.7%	53.2%	59.5%
llama 3-8b adapted	49.9%	64.0%	58.9%	60.5%

Table 6. English summarized comparisons between Llama 3 8B model pre and post adaptation where we see a 1% average improvement across all domains. There is a 1% drop in knowledge tasks which can be attributed to stability gap during domain adaptation

Model	XNLI Hindi	English downstream average
llama 2-13b base	37,44%	57.1%
llama 2-13b adapted	45.68%	55.3%

Table 7. Downstream performance comparison between Llama 2 13B pre and post adaptation on i) XNLI-Hindi, ii) English downstream task average of World Knowledge tasks, Commonsense reasoning tasks, and Misinformation and Bias tasks.

2018) where the goal is to predict textual entailment. We see 8% improvement compared to Llama 2 13B while in English there is 1.8% degradation as shown in 7.

5. Conclusion

We present an efficient recipe to significantly enhance capabilities of an English-dominant foundation LLM in another language. Our approach includes extending the vocabulary, applying a novel method for embedding initialization and alignment, and continually pre-training the foundation LLM on a bilingual data mix. We perform hyper-parameter optimization for batch size, learning rate schedule, and data mix ratio to ensure successful adaptation without experiencing "catastrophic forgetting". We successfully use this approach to enhance Arabic capability of Llama 2 base models, resulting in a state-of-the-art 70B Arabic base language model. Furthermore, we apply this approach to other languages such as Hindi and other foundation LLMs such as Llama 3 - while we briefly discuss these two studies but will leave the details for future work. We also intend to extend this approach to low resource languages and architecture such as depth-up scaling (Kim et al., 2024) and adapter layers methods as discussed in D.

References

Ahuja, K., Diddee, H., Hada, R., Ochieng, M., Ramesh, K., Jain, P., Nambi, A., Ganu, T., Segal, S., Ahmed, M., Bali, K., and Sitaram, S. MEGA: Multilingual evaluation of generative AI. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4232–4267, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.258. URL https://aclanthology.org/2023.emnlp-main.258.

Ali, M., Fromm, M., Thellmann, K., Rutmann, R., Lübbering, M., Leveling, J., Klug, K., Ebert, J., Doll, N., Buschhoff, J. S., Jain, C., Weber, A. A., Jurkschat, L., Abdelwahab, H., John, C., Suarez, P. O., Ostendorff, M., Weinbach, S., Sifa, R., Kesselheim, S., and Flores-Herr, N. Tokenizer choice for llm training: Negligible or crucial?, 2024.

Azerbayev, Z., Schoelkopf, H., Paster, K., Santos, M. D., McAleer, S., Jiang, A. Q., Deng, J., Biderman, S., and Welleck, S. Llemma: An open language model for mathematics, 2024.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Chen, Z., Cano, A. H., Romanou, A., Bonnet, A., Matoba, K., Salvi, F., Pagliardini, M., Fan, S., Köpf, A., Mohtashami, A., Sallinen, A., Sakhaeirad, A., Swamy, V., Krawczuk, I., Bayazit, D., Marmet, A., Montariol, S., Hartley, M.-A., Jaggi, M., and Bosselut, A. Meditron-70b: Scaling medical pretraining for large language models, 2023.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

Conneau, A., Rinott, R., Lample, G., Williams, A., Bowman, S. R., Schwenk, H., and Stoyanov, V. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

Csaki, Z., Pawakapan, P., Thakker, U., and Xu, Q. Efficiently adapting pretrained language models to new languages, 2023.

Da Dalt, S., Llop, J., Baucells, I., Pamies, M., Xu, Y., Gonzalez-Agirre, A., and Villegas, M. FLOR: On the effectiveness of language adaptation. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N. (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 7377–7388, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.650.

de Vries, W. and Nissim, M. As good as new. how to successfully recycle English GPT-2 to make models for other languages. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 836–846, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.74. URL https://aclanthology.org/2021.findings-acl.74.

Dey, N., Gosal, G., Zhiming, Chen, Khachane, H., Marshall, W., Pathria, R., Tom, M., and Hestness, J. Cerebras-gpt: Open compute-optimal language models trained on the cerebras wafer-scale cluster, 2023.

et. al., T. L. S. Bloom: A 176b-parameter open-access multilingual language model, 2023.

French, R. M. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999. ISSN 1364-6613. doi: https://doi.org/10.1016/S1364-6613(99)01294-2. URL https://www.sciencedirect.com/science/article/pii/S1364661399012942.

Fujii, K., Nakamura, T., Loem, M., Iida, H., Ohi, M., Hattori, K., Shota, H., Mizuki, S., Yokota, R., and Okazaki, N. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities, 2024.

Gadre, S. Y., Smyrnis, G., Shankar, V., Gururangan, S., Wortsman, M., Shao, R., Mercat, J., Fang, A., Li, J., Keh, S., Xin, R., Nezhurina, M., Vasiljevic, I., Jitsev, J., Dimakis, A. G., Ilharco, G., Song, S., Kollar, T., Carmon, Y., Dave, A., Heckel, R., Muennighoff, N., and Schmidt, L. Language models scale reliably with over-training and on downstream tasks, 2024.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., and Leahy, C. The pile: An 800gb dataset of diverse text for language modeling, 2020.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Guo, Y., Fu, J., Zhang, H., Zhao, D., and Shen, Y. Efficient continual pre-training by mitigating the stability gap, 2024. URL https://arxiv.org/abs/2406.14833.

Gupta, K., Thérien, B., Ibrahim, A., Richter, M. L., Anthony, Q., Belilovsky, E., Rish, I., and Lesort, T. Continual pre-training of large language models: How to (re)warm your model?, 2023.

Hardalov, M., Mihaylov, T., Zlatkova, D., Dinkov, Y., Koychev, I., and Nakov, P. EXAMS: A multi-subject high school examinations dataset for cross-lingual and multilingual question answering. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5427–5444, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.438. URL https://aclanthology.org/2020.emnlp-main.438.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034, 2015. doi: 10.1109/ICCV.2015.123.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models, 2022.

Huang, H., Yu, F., Zhu, J., Sun, X., Cheng, H., Song, D., Chen, Z., Alharthi, A., An, B., He, J., Liu, Z., Zhang, Z., Chen, J., Li, J., Wang, B., Zhang, L., Sun, R., Wan, X., Li, H., and Xu, J. Acegpt, localizing large language models in arabic, 2024.

Isik, B., Ponomareva, N., Hazimeh, H., Paparas, D., Vassilvitskii, S., and Koyejo, S. Scaling laws for downstream task performance of large language models, 2024.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023.

Kim, D., Park, C., Kim, S., Lee, W., Song, W., Kim, Y., Kim, H., Kim, Y., Lee, H., Kim, J., Ahn, C., Yang, S., Lee, S., Park, H., Gim, G., Cha, M., Lee, H., and Kim, S. Solar 10.7b: Scaling large language models with simple yet effective depth up-scaling, 2024. URL https://arxiv.org/abs/2312.15166.

Kusupati, A., Bhatt, G., Rege, A., Wallingford, M., Sinha, A., Ramanujan, V., Howard-Snyder, W., Chen, K., Kakade, S., Jain, P., and Farhadi, A. Matryoshka representation learning, 2024.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. RACE: Large-scale ReAding comprehension dataset from examinations. In Palmer, M., Hwa, R., and Riedel, S. (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods, 2021.

Loshchilov, I. and Hutter, F. Sgdr: stochastic gradient descent with warm restarts, 2017. URL https://doi.org/10.48550/arXiv.1608.03983.

Luo, Y., Yang, Z., Meng, F., Li, Y., Zhou, J., and Zhang, Y. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2024.

Marchisio, K., Lewis, P., Chen, Y., and Artetxe, M. Mini-model adaptation: Efficiently extending pretrained models to new languages via aligned shallow training. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5474–5490, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.338. URL https://aclanthology.org/2023.findings-acl.338.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Minixhofer, B., Paischer, F., and Rekabsaz, N. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V. (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3992–4006, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.293. URL https://aclanthology.org/2022.naacl-main.293.

Nangia, N., Vania, C., Bhalerao, R., and Bowman, S. R. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1953–1967, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.154. URL https://aclanthology.org/2020.emnlp-main.154.

Petrov, A., Malfa, E. L., Torr, P. H. S., and Bibi, A. Language model tokenizers introduce unfairness between languages, 2023.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. WINOGRANDE: an adversarial winograd schema challenge at scale. *CoRR*, abs/1907.10641, 2019. URL http://arxiv.org/abs/1907.10641.

Sap, M., Rashkin, H., Chen, D., Le Bras, R., and Choi, Y. Social IQa: Commonsense reasoning about social interactions. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454.

Sengupta, N., Sahu, S. K., Jia, B., Katipomu, S., Li, H., Koto, F., Afzal, O. M., Kamboj, S., Pandit, O., Pal, R., Pradhan, L., Mujahid, Z. M., Baali, M., Aji, A. F., Liu, Z., Hock, A., Feldman, A., Lee, J., Jackson, A., Nakov, P., Baldwin, T., and Xing, E. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models, 2023a.

Sengupta, N., Sahu, S. K., Jia, B., Katipomu, S., Li, H., Koto, F., Marshall, W., Gosal, G., Liu, C., Chen, Z., Afzal, O. M., Kamboj, S., Pandit, O., Pal, R., Pradhan, L., Mujahid, Z. M., Baali, M., Han, X., Bsharat, S. M., Aji, A. F., Shen, Z., Liu, Z., Vassilieva, N., Hestness, J., Hock, A., Feldman, A., Lee, J., Jackson, A., Ren, H. X., Nakov, P., Baldwin, T., and Xing, E. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models, 2023b.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In Erk, K. and Smith, N. A. (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

Tay, Y., Dehghani, M., Rao, J., Fedus, W., Abnar, S., Chung, H. W., Narang, S., Yogatama, D., Vaswani, A., and Metzler, D. Scale efficiently: Insights from pre-training and fine-tuning transformers, 2022.

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

Wu, C., Gan, Y., Ge, Y., Lu, Z., Wang, J., Feng, Y., Shan, Y., and Luo, P. Llama pro: Progressive llama with block expansion, 2024.

Yang, T., Li, C., Zhu, Y., Sun, Y., Huang, F., and Lin, C. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

Zhao, J., Zhang, Z., Gao, L., Zhang, Q., Gui, T., and Huang, X. Llama beyond english: An empirical study on language capability transfer, 2024.

A. Tokenizer ablations

We experimented with one more tokenizer variant, *Llama3-replace5* in addition to *Llama 2-extend100* and *Llama3-replace30*. Here we replace only 5% of the Llama 2 tokenizaer vocabulary with that of MLV2's most frequent Arabic tokens. Experiment design:

- Continually pre-train a monolingual LLM on native + target language (Arabic) mix tokenized with the new vocabulary.
- Use the same hyperparameters across the ablations.
- Fix the raw text corpus for each language – this will ensure fairness as the total information/bytes are fixed.
- Select the size of raw text corpus for each language such that when tokenized, by a monolingual tokenizer in the respective language, the total tokens are in the same range.
- **Pre-trained base model:** Llama 2-7B
- **Datasets:** Pile and AraV5.
- **Embedding initialization:** mean or subword-mean (discussed in the next section).
- Learning rate of $1.5e-4$ and batch size of 4M tokens.

We take the raw corpus (bytes) of Pile and AraV5 such that when tokenized by the MLV2 Arabic tokenizer, the number of tokens for Pile and AraV5 are same. In table 8 we show extrinsic evaluation of different bilingual tokenizers using either *extend* or *replace* new token injection/merge schemes. Overall *Llama 2 base* performed the best but it had $> 3.5x$ more Arabic tokens compared to *Llama 2-extend100* which is not compute efficient. But if we compare isoToken in terms of Arabic, *Llama 2-extend100* performs the best. Another factor is the tokens per parameter (TPP), with *Llama 2-replace5* and *Llama 2-base* the TPP is much larger, and as we're going to be training for higher TPPs, using a high fertility tokenizer would drift far from the pareto-frontier or in other words there will be TPP degradation. Another thing to consider here is inference cost of high fertility tokenizers as more tokens for the given text mean limited context for the model, more memory requirements.

B. Cross Entropy Loss Correction

When comparing cross-entropy loss between models trained with data tokenized by different tokenizers, we need to apply correction or normalization to the losses. This normalization is required because cross-entropy's units of measurement are `nats/tokens`, and therefore, the definition of a token becomes very important. Depending on the size of the vocabulary and type of tokenizer, the information represented per token varies. The loss correction factor to compare cross-entropy loss between two models is then the ratio of the number of tokens in the validation sets for each model.

C. Embedding initialization

Here we discuss the ablations that we performed with different embeddings initialization methods as discussed in the main body. We also ablated an additional initialization method which we refer to as *Subword Mean*. Following are the initialization methods under consideration:

- **Mean**: Initialize all the new tokens' embeddings with the mean of source language token embeddings.
- **Subword Mean**: For a newly added Arabic token, tokenize it using base Llama tokenizer and use the mean of the token embeddings ot the sequence of tokens.
- **Semantic similarity search based**: This method was introduced in Wechsel multilingual initialization work where a pre-trained embeddings like Fasttext or OpenAI embeddings are used.
- **Projection based:** Use least squared to Learn a transformation matrix from a learned embedding space (MLV2) to the Llama token embeddings space using the overlapping tokens. Then apply this transformation to the newly added tokens from MLV2 vocab to Llama vocab.

Note that for LLMs with untied embeddings and unembeddings, the new tokens embeddings (or unembeddings) are initialized independently using the above methods.

In table 9 we compare different initialization methods for Llama2 7B continual pre-training on AraV5 and Pile mix tokenized with *Llama2-extend100*.

D. Block expansion adapter approach for multilingual models

Following the work outlined in (Wu et al., 2024) we leverage the block expansion approach for multilingual models, making it highly effective for language adaptation. By adding and fine-tuning additional Transformer blocks initialized to identity mappings, the model can integrate new domain-specific knowledge without forgetting previous information. Although, the techniques described in the original paper focus on code and math, we were able to successfully adapt the approach for our experiments with English and Arabic. We initialized our base model with Llama-2 7B and expanded the number of blocks from 32 to 40 using an interleaved

variant	merge	Emb. initialization	Vocab size	Total Tokens	Arabic Tokens	English Tokens	Pile loss	AraV5 loss
Llama 2 base	N/A	N/A	32000	87.5B	67.31B	20.22B	1.508	0.637
Llama 2-replace5	replace	Mean	32000	51B	30.7B	20.3B	1.504	0.6493
Llama 2-replace5	replace	Subword Mean	32000	51B	30.7B	20.3B	1.508	0.646
Llama 2-replace30	replace	Subword Mean	32000	43.7B	22.76B	20.94B	1.516	0.6439
Llama 2-extend100	extend	Mean	64000	39.7B	19.35B	20.35B	1.499	0.6591
Llama 2-extend100	extend	Subword Mean	64000	39.7B	19.35B	20.35B	1.499	0.6539

Table 8. Extrinsic evaluation of different bilingual tokenizers using either *extend* or *replace* new token injection/merge schemes. Due to varying fertility in Arabic we can see that tokenizer with smaller number of Arabic tokens in its vocabulary has a higher fertility and thus a higher token count. So this is not an iso-token comparison.

Emb. initialization	Total Tokens	Arabic Tokens	English Tokens	Pile loss	AraV5 loss
Mean	39.7B	19.35BB	20.35B	1.4995	0.6591
Subword Mean	39.7B	19.35BB	20.35B	1.4994	0.653
Wechsel $k = 5$	39.7B	19.35BB	20.35B	1.4988	0.64898
Wechsel $k = 10$	39.7B	19.35BB	20.35B	1.4999	0.656
MLV2 Projection	39.7B	19.35BB	20.35B	1.5013	0.64857

Table 9. Comparison of different embedding initialization techniques in terms of upstream eval loss of Llama2 7B when trained on Pile and AraV5 mic for 39.7 billion tokens.

Tokenizer variant	Emb. initialization	First step train loss
Llama 2 replace5	Random	16.31
Llama 2 replace5	Mean	9.15
Llama 2 extend5	Mean	9.43
Llama 2 extend100	Mean	7.22
Llama 2 extend100	Subword Mean	5.47
Llama 2 extend100	Wechsel $k = 5$	5.49
Llama 2 extend100	MLV2 projection	4.8

Table 10. Comparison of different embedding initialization techniques across multiple tokenizers.

approach. In our experiments for language adaptation, we found that an optimal data mix of 1:9(En:Ar) yielded the best results (in downstream 0 shot tasks in both English and Arabic) relative to adapting the newly added layers only on domain specific data. In both experiments we trained on a total of 67B tokens in Arabic in order to maintain the same token count for the appropriate comparison. Our results show that the block-expansion approach is a strong candidate for language adaptation with a faster time to train and lower training costs. In the future, this work could expand to other types of models(like MoE models) and modalities and would be interesting to analyse the impact on overall accuracy in downstream tasks For language adaptation with block expansion [sectionD], we experiment with different number of adapter layers. We find that the optimal adapter layer is 25% of the existing layer. Similarly, 960 is the optimal batch size. Table 11 summarizes our results using the above approach for language adaptation at the LLama 2 7B scale

E. Fine-tuning

Upstream loss is typically assumed to indicate downstream performance [(Isik et al., 2024), (Gadre et al., 2024)]. In order to verify performance on downstream tasks in the adapted domain, we fine-tune both pre-trained and adapted pre-trained models. Instruction fine-tuning allows us to assess both performance and generation quality, which may not always match upstream performance (Tay et al., 2022).

The data used is an extended fine-tuning dataset following from (Sengupta et al., 2023b). We add additional data in English and Arabic, focusing on bilingual examples and quality for language and style adaptation. In total, our instruction fine-tuning dataset contains approximately 10 million English and 4 million Arabic examples in both single-turn and multi-turn settings

We fine-tuned for 3 epochs with a standard linear learning rate decay. Instead of padding, examples are packed together up to the sequence length and separated with EOS tokens, increasing training efficiency by up to 8 times. As in Jais, we calculate loss on answer tokens only.

We observe that downstream performance of the fine-tuned model trained on top of the pre-trained model is lower than that of the fine-tuned model trained after domain adaptation. It suggests that, similar to the findings in (Isik et al., 2024), downstream task performance after fine-tuning is highly dependent on the alignment between pre-training data and downstream tasks which is improved through adaptation.

Model	Data Mix(En:Ar)	Block expansion %	Arabic tokens	Arabic downstream eval acc%	English downstream eval acc%
Llama 2 7B	N/A	N/A	N/A	38.34	54.68
Llama 2 7B	0:1	25	67B	42.52	55.69
Llama 2 7B	1:9	25	67B	43.16	57.80

Table 11. Evaluation of block expansion adapter approach with data mixes across various downstream evaluation tasks. Arabic tasks are evaluated across Knowledge, Commonsense reasoning and Misinformation, bias. English tasks are evaluated for Commonsense reasoning

F. Hardware setup

The training runs were conducted on two Condor Galaxy supercomputers, each equipped with 64 Cerebras CS-2 Wafer-Scale Engines (WSE-2). Each CS-2 features 40 GB of SRAM and achieves a peak throughput of 7.5 PetaFLOP/s in half precision, providing a total of 960 PetaFLOP/s in half precision across both supercomputers. Utilizing the weight streaming mode of the Cerebras software stack, the Condor Galaxy supercomputers can flexibly schedule multiple jobs based on hardware resource requirements and priority. The number of CS-2s allocated to a job can be dynamically adjusted during training, with performance scaling linearly up to 64 CS-2s per job. This scalability is facilitated by the Cerebras software stack's use of pure data parallelism to distribute the workload across multiple CS-2s. Jobs are managed by a priority queue system, ensuring efficient allocation of computational resources.

G. Downstream Tasks

In 12 and 13 we present downstream task performance breakdown on Arabic and English downstream tasks, respectively. We draw the comparison between Llama 2 series of Arabic adapted models compared against base Llama 2 models and open source state of the art Arabic models. *Llama 2 13B*, *Llama 2 13B adapted*.

Tables 14 and 15 show the downstream task performance breakdown on Arabic and English downstream tasks, respectively, for *Llama 3 8B* base and *Llama 3 8B adapted*.

	Knowledge		Commonsense Reasoning						Misinformation, bias		
model	mmlu (acc_norm)	exams	Hellaswag	PIQA	BoolQ(acc)	SIQA	ARC Challenge	Openbook QA	TruthfulQA	CrowS-Pairs(pct_stereotype)	Average
llama 2-70b adapted	37.7%	39.1%	61.6%	68.2%	66.9%	41.4%	41.2%	33.2%	45.6%	57.2%	49.2%
llama 2-13b adapted	30.6%	37.7%	54.9%	67.1%	64.5%	40.6%	36.1%	32.0%	43.6%	54.0%	46.1%
llama 2-7b adapted	28.7%	39.0%	48.0%	62.8%	63.9%	38.5%	32.0%	31.4%	43.9%	54.2%	43.51%
llama 2-70b base	30.2%	32.6%	41.2%	54.6%	64.2%	35.2%	30.5%	31.4%	47.0%	50.9%	41.8%
llama 2-13b base	28.4%	30.4%	34.3%	52.9%	63.8%	36.4%	24.3%	30.0%	45.5%	49.9%	39.6%
llama 2-7b base	27.8%	26.7%	32.3%	50.0%	63.8%	35.6%	25.0%	29.0%	46.7%	48.3%	38.5%
Acegpt-13b	29.9%	34.7%	45.6%	60.3%	63.2%	38.1%	32.8%	32.2%	45.1%	56.4%	43.8%
Jais 30b	34.0%	42.0%	60.4%	69.0%	67.7%	42.2%	39.2%	33.8%	45.1%	57.3%	49.1%

Table 12. Detailed break down of **Knowledge**, **Commonsense Reasoning** and **Misinformation, bias** downstream Arabic task performance of Llama 2 adapted models, and comparison against base Llama 2 and state of the art Arabic open source models

	Knowledge		Commonsense Reasoning							Misinformation, Bias		
model	mmlu	race	Hellaswag	PIQA	BoolQ	SIQA	ARC Challenge	Openbook QA	Winogrande	TruthfulQA	CrowS-Pairs(pct_stereotype)	Average
llama 2-70b adapted	52.2%	39.6%	82.0%	81.5%	82.8%	46.2%	52.1%	45.8%	75.9%	43.8%	68.2%	60.9%
llama 2-13b adapted	37.3%	39.5%	76.5%	78.6%	77.8%	44.6%	45.9%	44.4%	71.4%	34.6%	64.0%	55.9%
llama 2-7b adapted	33.9%	38.6%	74.0%		75.4%	44.4%	42.2%	43.6%	67.3%	37.6%	65.7%	52.3%
llama 2-70b base	55.6%	42.0%	80.8%	81.0%	76.8%	42.6%	48.0%	44.4%	76.9%	44.5%	69.6%	60.2%
llama 2-13b base	34.9%	40.8%	76.6%	79.1%	69.0%	44.9%	44.3%	42.0%	69.6%	37.6%	69.8%	55.3%
llama 2-7b base	32.0%	40.1%	73.0%	77.0%	71.1%	42.7%	40.5%	40.8%	67.2%	39.6%	71.1%	54.1%
Acegpt-13b	34.6%	39.7%	77.0%	79.6%	77.6%	45.7%	44.2%	40.0%	70.1%	39.4%	73.7%	56.5%
Jais 30b	42.3%	40.3%	79.1%	80.5%	80.9%	49.3%	48.4%	43.2%	70.6%	40.3%	72.3%	58.8%

Table 13. Detailed break down of **Knowledge**, **Commonsense Reasoning** and **Misinformation, bias** downstream English task performance of Llama 2 adapted models, and comparison against base Llama 2 and state of the art Arabic open source models

	Knowledge		Commonsense Reasoning						Misinformation, bias		
model	mmlu (acc_norm)	exams	Hellaswag	PIQA	BoolQ(acc)	SIQA	ARC Challenge	Openbook QA	TruthfulQA	CrowS-Pairs(pct_stereotype)	Average
llama 3-8b	31.0%	34.7%	45.2%	57.7%	66.4%	37.5%	34.6%	30.0%	48.3%	46.4%	43.2%
llama 3-8b adapted	36.0%	41.5%	51.8%	64.5%	71.4%	41.2%	35.8%	31.0%	52.1%	54.4%	47.9%

Table 14. Detailed break down of **Knowledge**, **Commonsense Reasoning** and **Misinformation, bias** downstream Arabic task performance of Llama 3 8B adapted model, and comparison against base Llama 3 8B

	Knowledge		Commonsense Reasoning							Misinformation, Bias		
model	mmlu	race	Hellaswag	PIQA	BoolQ	SIQA	ARC Challenge	Openbook QA	Winogrande	TruthfulQA	CrowS-Pairs(pct_stereotype)	Average
llama 3-8b	62.0%	40.1%	79.2%	80.9%	81.2%	33.1%	53.4%	45.0%	72.8%	43.9%	62.3%	59.5%
llama 3-8b adapted	60.6%	39.0%	79.1%	80.5%	82.3%	33.2%	53.5%	46.2%	73.3%	51.1%	66.7%	60.5%

Table 15. Detailed break down of **Knowledge**, **Commonsense Reasoning** and **Misinformation, bias** downstream English task performance of Llama 3 8B adapted model, and comparison against base Llama 3 8B