# ⚓ Sailor: Open Language Models for South-East Asia

**Longxu Dou**[1*] **Qian Liu**[1*] **Guangtao Zeng**[2] **Jia Guo**[1] **Jiahui Zhou**[1]
**Xin Mao**[1] **Ziqi Jin**[2] **Wei Lu**[2] **Min Lin**[1]

[1]Sea AI Lab, Singapore [2]SUTD, Singapore

## Abstract

We present Sailor, a family of open language models ranging from 0.5B to 14B parameters, tailored for South-East Asian (SEA) languages. From Qwen1.5, Sailor models accept 200B to 400B tokens during continual pre-training, primarily covering the languages of English, Chinese, Vietnamese, Thai, Indonesian, Malay, and Lao. The training leverages several techniques, including BPE dropout for improving the model robustness, aggressive data cleaning and deduplication, and small proxy models to optimize the data mixture. Experimental results on four typical tasks indicate that Sailor models demonstrate strong performance across different benchmarks, including commonsense reasoning, question answering, reading comprehension and examination. We share our insights to spark a wider interest in developing large language models for multilingual use cases. Our demo can be found at `https://hf.co/spaces/sail/Sailor-14B-Chat`.

## 1 Introduction

Large language models (LLMs) have seen remarkable improvements recently, driven by the rapid growth of Internet data (Rana, 2010) and advances in pre-training techniques. However, mainstream LLMs (Touvron et al., 2023a; AI et al., 2024; Bai et al., 2023) primarily rely on English data for training. For example, 89.70% of the training data of Llama-2 is English (Touvron et al., 2023b). Consequently, these English-centric LLMs often struggle to achieve comparable performance across other languages (e.g., Thai), due to their inadequate exposure to those languages during pre-training.

In this paper, we aim to develop the LLMs that perform well across the South-East Asia (SEA) region, encompassing a range of languages that include English, Chinese, Vietnamese, Thai, Indonesian, Malay, and Lao. To cater to varying needs, **we release both base model and chat model in five variant size** (0.5B, 1.8B, 4B, 7B and 14B)[1], offering greater flexibility. Additionally, **we open source all of our data cleaning and deduplication pipeline**[2] that turns out to be extremely important for the quality of LLMs, especially in the scenario of continual pre-training.

Besides the open models, **we explore several techniques in a fully transparent manner** to accelerate the development of multilingual LLMs, which encompasses three main areas of investigation. First, we employ small-scale models as proxies to optimize hyperparameters for continual pre-training, focusing on learning rates and data mixture ratios from diverse sources. Second, we examine the efficacy of various data processing techniques, including the merging of adjacent short examples, as well as document-level and word-level code-switching. Finally, we address tokenization challenges by investigating the use of BPE dropout (Provilkov et al., 2020) to improve the robustness of LLMs.

With exploring the above techniques, **we summarize the key insights for multilingual LLM continual pre-training**, as illustrated in Figure 1: (1) Language models struggle with multiple languages, and continual pre-training presents an opportunity to improve specific language capabilities. (2) Code-switching techniques can be beneficial in multilingual scenarios, improving the ability to handle language mixing. (3) Language models are sensitive to subword segmentation, and techniques like BPE dropout can improve model robustness. (4) Even available high-quality multilingual corpora may still require further data deduplication and cleaning. (5) Simulation experiments on smaller models can provide insights into performance trends for large-scale experiments.

*The first two authors contributed equally. Contact doulx@sea.com for more information.

[1]`https://hf.co/models?search=sail-Sailor`
[2]`https://github.com/sail-sg/sailcraft`



*Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing System Demonstrations*, pages 424–435
November 12-16, 2024 

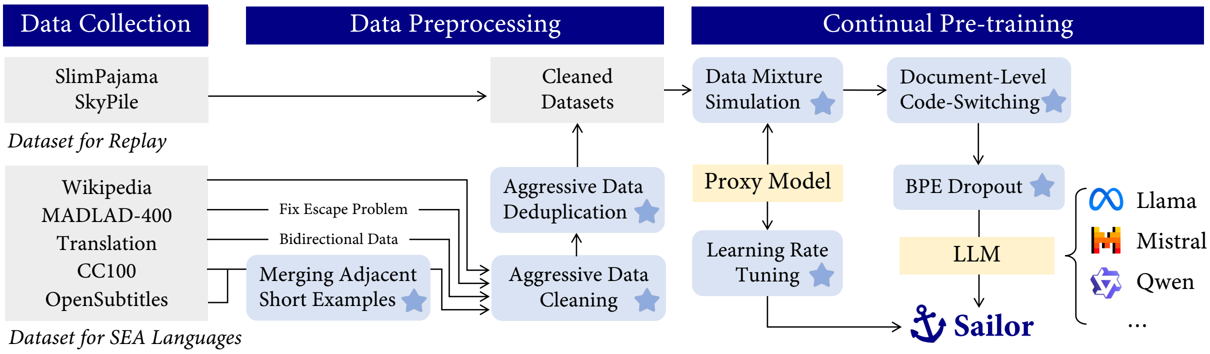



Figure 1: The pipeline of building Sailor, with key insights marked by blue stars.

## 2 Continue Pre-training for Base Model

A crucial aspect of continual pre-training is meticulous data processing and the selection of a suitable LLM as the foundation. This section outlines our data processing pipeline, model selection criteria, and implementation details.

### 2.1 Data Processing

**Data Sourcing** (1) For English and Chinese, we choose SlimPajama (Soboleva et al., 2023) and SkyPile (Wei et al., 2023) as replay data. (2) For SEA languages, we choose CC100 (Wenzek et al., 2020), MADLAD-400 (Kudugunta et al., 2023) and Wikipedia[3] as multilingual dataset. (3) To enrich the SEA corpus, we collect the Malay, Indonesian, Thai and Vietnamese subtitles from the OPUS OpenSubtitles category[4]. (4) To improve the document-level code-switching, we curate a selection of English-SEA language translation pairs (e.g., TED2020 talks) from OPUS project[5].

**Data Cleaning** The data quality is crucial for model pre-training. We find that the publicly available multilingual datasets (e.g., CC100 and MADLAD-400) could be further cleaned and deduplicated. To improve the data cleaning process for SEA languages specifically, we expanded the list of filtering words, trained new filtering models, and implemented a more aggressive deduplication strategy. Eventually, we extracted 61.19% of data for SEA languages from public datasets, and constructed the final SailCraft dataset. The specific removal rates are shown in Figure 2.

[3]https://huggingface.co/datasets/wikimedia/wikipedia
[4]https://opus.nlpl.eu/OpenSubtitles-v2018.php
[5]https://opus.nlpl.eu/

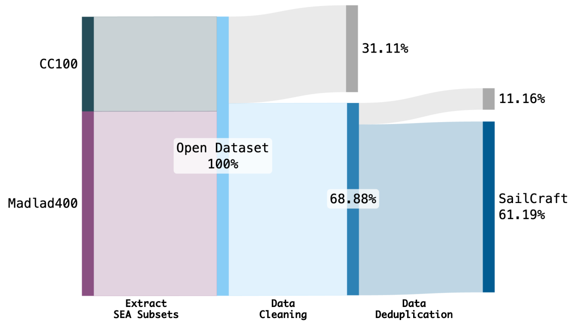



Figure 2: This forms the **SailCraft** dataset, used to train the **Sailor** models. The reported removal rate (grey) is with respect to each previous stage, and the kept rate (colored) demonstrates the overall rate.

**Data Mixture** We aim to develop an SEA tailored LLM but kept the original capability (e.g., English) simultaneously, requiring the balanced representation across all target languages. To achieve this, we develop the algorithm RegMix that determines the appropriate weights for various languages during pre-training. As depicted in Figure 3, we begin by training a set of proxy models (e.g., 64 in total here) on a variety of data mixtures for a limited number of training steps (e.g., 1000 steps). We then fit a linear regression model, using the data mixture as the input feature and the joint loss considering all languages as the target[6]. With this model, we can perform numerous simulation experiments (e.g., 1,000,000) on randomly sampled data mixtures to explore the vast array of possibilities within seconds. The linear model then guides us in selecting the combination that yields the lowest predicted joint loss. Once this data mixture has been optimized, it can be directly applied to large-scale training. More details and findings could be found in the RegMix paper (Liu et al., 2024) .

[6]We use the product of individual losses as the joint loss.

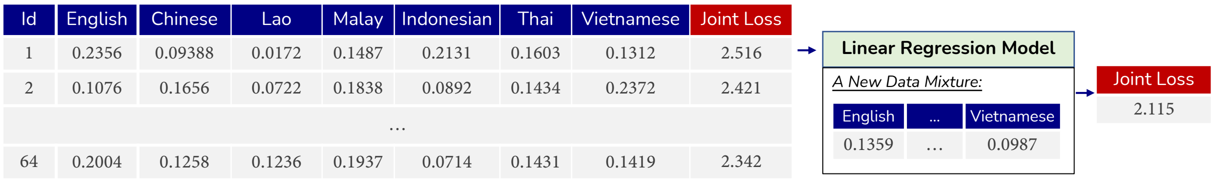



Figure 3: We employ the experimental results from proxy models across a variety of data mixtures (e.g., 64 distinct data mixture here) to fit a linear regression model. The model is then utilized to predict the validation loss of simulate numerous random data mixtures, enabling us to identify the most effective data mixture for optimizing joint loss. Subsequently, the best data mixture is applied to large-scale training.

| Language | Source | Tokens (B) | Epoch |
|---|---|---|---|
| EN | SlimPajama | 37.20 | 0.06 |
| ZH | SkyPile | 22.64 | 0.15 |
| LO | CC100 | 0.03 | 0.97 |
| | MADLAD | 0.31 | 0.97 |
| MY | CC100 | 2.02 | 1.34 |
| | MADLAD | 5.54 | 1.54 |
| | OpenSubtitles | 0.04 | 1.07 |
| | Wikipedia | 0.17 | 1.32 |
| ID | CC100 | 23.72 | 0.90 |
| | MADLAD | 25.62 | 0.66 |
| | OpenSubtitles | 0.24 | 1.07 |
| | Wikipedia | 0.45 | 1.32 |
| | Translation | 0.50 | 1.16 |
| TH | CC100 | 3.00 | 1.28 |
| | MADLAD | 32.07 | 1.35 |
| | OpenSubtitles | 0.13 | 1.01 |
| | Wikipedia | 0.28 | 1.32 |
| | Translation | 0.34 | 1.14 |
| VI | CC100 | 14.25 | 0.82 |
| | MADLAD | 26.16 | 0.44 |
| | OpenSubtitles | 0.05 | 1.08 |
| | Wikipedia | 0.50 | 1.32 |
| | Translation | 0.43 | 1.20 |

Table 1: The data composition of the final corpus.

**Data Composition** To achieve better mixture performance, we further incorporate the data source factor into RegMix implementation. This means we treat each language from every source as a distinct dataset and try to optimize the data mixture of these datasets. Empirically, we adopt Qwen1.5-0.5B model as the proxy model, then apply it for optimizing the data mixture for continual pre-training process across all model sizes. The effective tokens and equivalent epochs in SailCraft are documented in Table 1. We could observe that CC100 exhibits a relative advantage over MADLAD-400, in terms of quality or diversity, particularly for Indonesian and Vietnamese. The final pre-training corpus is composed of approximately 200B tokens, integrating both SEA tokens and replay tokens.

## 2.2 Model Selection

We select Qwen1.5 family models as the foundation for Sailor models due to their extensive vocabulary (151K tokens) and multilingual-friendly byte distribution, which offer significant potential for future enhancements (Tao et al., 2024). We adopt most of the pre-training settings and model architectures from Qwen1.5 (Bai et al., 2023). It follows the standard transformer architecture (Vaswani et al., 2017), adopts the pre-normalization with RMSNorm (Jiang et al., 2023b), SwiGLU activation (Shazeer, 2020) and rotary positional embeddings (Su et al., 2022).

## 2.3 Implementation Details

**Codebase** To balance the training efficiency and debugging convenience, we leverage two codebases for different size model. For relatively large models (i.e., 4B, 7B, 14B), we utilize Megatron-LM[7] (Shoeybi et al., 2019), which supports tensor parallel and pipeline parallel to maximize the model flops utilization (MFU) of NVIDIA GPUs. For relatively small models (i.e., 0.5B and 1.8B), we employ the TinyLlama (Zhang et al., 2024) codebase[8], which follows a compact structure and allows easy modifications for diverse purposes.

**Hyper-parameters** We employ a batch size of 4M tokens and a learning rate of 1e-4. After a 500-step warmup period, the learning rate is maintained at a constant level following Hu et al. (2024). This scheduling strategy encourages more transferable conclusions from simulations and allows for easier recovery from interrupted training sessions. Sailor models typically train on 200B tokens (one epoch of SailCraft corpus), except for Sailor-0.5B which trains on 400B tokens (two epochs). We train models with BFloat16 mixed precision to balance the training efficiency and stability.

[7] https://github.com/epfLLM/Megatron-LLM
[8] https://github.com/jzhang38/TinyLlama

## 3 Post-training for Chat Model

### 3.1 Supervised Fine-tuning

**Training Dataset** The instruction tuning corpus includes four open instruction tuning datasets: Aya Collection (Singh et al., 2024), Aya Dataset (Singh et al., 2024), SlimOrca (Lian et al., 2023) and UltraChat (Ding et al., 2023)[9]. For Aya Collection and Aya Dataset, we select the English, Chinese, and SEA language subsets for fine-tuning. For SlimOrca and UltraChat, we use NLLB (Costa-jussà et al., 2022) to translate them from English into SEA languages. Additionally, we extract the system prompts from SlimOrca, and translate them into SEA languages to augment the other three datasets. The final number of tokens used for fine-tuning is approximately 5.6B.

**Training Details** During the SFT training stage, following Llama (Touvron et al., 2023c), we mask out the tokens loss of system prompt and user tokens, only optimizing the assistant tokens. That is, we restrict backpropagation to only the answer tokens. For 0.5B model to 7B model, we utilize a training batch size of 4M and a learning rate of 1e-5. For 14B model, we utilize a training batch size of 1M and a learning rate of 2e-6. For each model size, we train the SFT dataset for three epochs.

### 3.2 Preference Optimization

**Training Dataset** Due to the high cost of constructing preference data for Southeast Asian languages, we use NLLB 3.3B model (Costa-jussà et al., 2022) to translate the UltraFeedback dataset (Cui et al., 2023) into Thai, Vietnamese, Malay, and Indonesian. After filtering out samples with excessively low perplexity, the remaining preference data is used for preference optimization.

**Training Details** During the RLHF stage, we use DPO (Rafailov et al., 2023) to align the model with human preferences and improve generation quality. During the training, we set the learning rate to 5e-7, $\beta$ to 0.05, and the batch size to 128.

## 4 Evaluation

In this section, we evaluate Sailor base models and other baseline models, on four typical NLP tasks across three main SEA languages (i.e., Indonesian, Thai, Vietnamese).

### 4.1 Benchmark

**Question Answering** XQuAD (Artetxe et al., 2020) (for Thai and Vietnamese) and TydiQA (Clark et al., 2020) (for Indonesian) are question-answering benchmarks. XQuAD contains 1,190 translated question-answer pairs from SQuAD v1.1's development set (Rajpurkar et al., 2016). TydiQA includes 204,000 pairs with original language data and human-written questions.

**Commonsense Reasoning** XCOPA (Ponti et al., 2020) (Indonesian, Thai, and Vietnamese) presents premises with two choices. Models must select the option that best represents either the cause or effect of the given event.

**Reading Comprehension** BELEBELE (Bandarkar et al., 2023) is a multilingual reading comprehension dataset covering 122 languages. We use its Indonesian, Thai, and Vietnamese subsets for evaluation. Each question includes a context paragraph and four answer choices.

**Examination** The M3Exam dataset (Zhang et al., 2023) (Javanese, Thai, Vietnamese) is a multilingual exam benchmark collected from official school tests used in nine countries[10].

### 4.2 Evaluation Protocol

We employed the evaluation platform OpenCompass (Contributors, 2023) to build up our evaluation code[11]. The performance of all models is assessed based on the 3-shot Exact Match (EM) and F1 performance, with prompts provided in native languages (e.g., Indonesian task description for Indonesian tasks).

For XCOPA and BELEBELE evaluations, we adopt the approach used by OpenCompass and the Eleuther AI evaluation framework (Gao et al., 2023) on the HellaSwag benchmark (Zellers et al., 2019). We reformulate these tasks as the continuation writing task. Each potential answer is appended to the given input or question, with the lowest perplexity score determining the prediction. As for M3Exam evaluation, we employ the official method described by Zhang et al. (2023). This approach involves directly prompting language models to generate the correct option ID when presented with a question and its corresponding choices.

[9]We employ the filtered version of the UltraChat: https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k.

[10]Note that we chose its Javanese subset since the Indonesian version has yet to be released when submitting this paper.

[11]https://github.com/sail-sg/sailor-llm.

| 3-shot (EM) | QA | Commonsense | RC | Examination | Total Score |
|---|---|---|---|---|---|
| Llama-2-7B | 44.75 | 59.60 | 36.52 | 26.42 | 167.29 |
| Mistral-7B-v0.1 | 55.25 | 60.40 | 39.00 | 34.71 | 189.35 |
| Sea-Lion-7B | 45.35 | 63.07 | 36.30 | 24.12 | 168.83 |
| SeaLLM-7B-Hybrid | 49.98 | 65.80 | 41.30 | 29.77 | 186.84 |
| SeaLLM-7B-v2 | 44.45 | 61.80 | 42.15 | 38.63 | 187.02 |
| Qwen1.5-0.5B | 18.25 | 52.33 | 29.00 | 24.53 | 124.12 |
| Sailor-0.5B | 22.47 | 55.73 | 31.81 | 24.75 | 134.76 (+10.65) |
| Qwen1.5-1.8B | 28.71 | 52.53 | 31.15 | 28.78 | 141.18 |
| Sailor-1.8B | 35.94 | 60.40 | 34.81 | 27.07 | 158.23 (+17.05) |
| Qwen1.5-4B | 42.02 | 55.40 | 34.74 | 32.16 | 164.32 |
| Sailor-4B | 49.48 | 63.60 | 38.78 | 29.31 | 181.17 (+16.85) |
| Qwen1.5-7B | 55.86 | 60.87 | 41.07 | 40.04 | 197.84 |
| Sailor-7B | 57.41 | 67.80 | 43.74 | 42.05 | 211.00 (+13.16) |
| Qwen1.5-14B | 57.76 | 68.73 | 42.66 | 45.56 | 214.72 |
| Sailor-14B | 55.40 | 74.80 | 45.19 | 49.55 | 224.94 (+10.22) |

Table 2: Each model's average score across three SEA languages for various tasks. The total score is the sum of scores from four tasks, representing the model's comprehensive performance. We also highlight the improvement of Sailor models over the Qwen1.5 models (in parentheses). Detailed experimental results can be found in Appendix A.

### 4.3 Baseline Setup

We choose three types of baseline models:

**General LLMs** general multilingual models, whose training corpus cater to multilingual tokens, but mainly focus on Western languages. It includes Llama-2 (Touvron et al., 2023b), Mistral (Jiang et al., 2023a), Qwen1.5 (Bai et al., 2023).

**SEA-specific LLMs by continual pretraining** train the General LLMs with SEA corpus, including VinaLLaMA (Nguyen et al., 2023a), SeaLLM (Nguyen et al., 2023b) and Typhoon (Pipatanakul et al., 2023).

**SEA-specific LLMs by training from scratch** training corpus consists of a significant number of SEA tokens and employ SEA friendly tokenizer, including Sea-Lion (AI Singapore, 2023).

### 4.4 Experimental Results

Experimental results shown in Table 2 indicate that Sailor models obviously outperform the baseline models in all variant sizes. Notably, we omit the results of VinaLLaMA and Typhoon, since they are solely optimized for one SEA language and incur performance degeneration in other languages.

We could observe that: (1) Sailors exceed the Qwen1.5 baseline model, highlighting the success of continual pre-training; (2) Sailors surpass other SEA-specific models, demonstrating the importance of careful data cleaning and data deduplication.

## 5 Insights

During Sailor development, we perform ablation studies on small LMs to understand the impact of various strategies[12]. We then apply the key insights gained from these studies to improve LLM. All techniques are listed in Table 3.

### 5.1 Data

**Merging Adjacent Short Examples** While deduplication improves data efficiency, it can disrupt contextual relevance. To address this, we randomly combine adjacent examples before global shuffling. This method works because deduplicated paragraphs retain their original order, allowing context reconstruction. We also apply this approach to inherently short-sentence sources like subtitles.

**Code-Switching** Code-switching involves using multiple languages within one context. We explore two types: document-level and word-level. Document-level mixing combines texts from various languages during pre-training. Word-level switching replaces 10% of words in SEA language documents with English equivalents. Our experiments with TinyLlama show that document-level switching outperforms word-level or combined approaches. Thus, we only use document-level switching in continual pre-training.

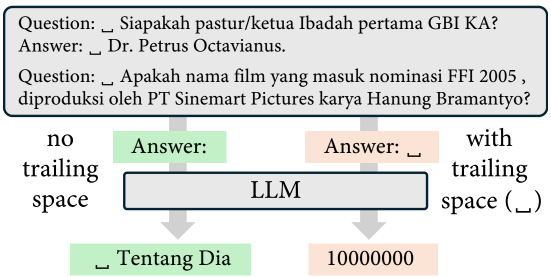
[12]Most of the experimental results are obtained from three series of models: our internal 120M model trained on 20B English tokens using SlimPajama (Soboleva et al., 2023), the TinyLlama 1.1B model (Zhang et al., 2024), and the Qwen1.5-0.5B model (Bai et al., 2023).

| Technique | Stage | Used | Note |
|---|---|---|---|
| Merging Adjacent Short Examples | Data | Yes | Improve Performance |
| Document-Level Code-Switching | Data | Yes | Improve Performance |
| Word-Level Code-Switching | Data | No | Marginal Effect w. Document-Level |
| Aggressive Data Deduplication | Data | Yes | Improve Performance |
| Aggressive Data Cleaning | Data | Yes | Improve Performance |
| BPE Dropout | Tokenization | Yes | Improve Robustness |
| Vocabulary Expansion | Tokenization | No | Challenging to Apply |
| Learning Rate Tuning | Training | Yes | Accelerate the Training |
| Data Mixture Simulation | Training | Yes | Balance Different Languages |

Table 3: The techniques we mainly consider during our development.

**Aggressive Data Cleaning and Deduplication** Even though we started with well-curated open datasets, e.g., MADLAD-400 clean set (Kudugunta et al., 2023), we still further removed 31.11% in data cleaning and 11.16% in data deduplication. By extensively filtering out noisy, harmful, and duplicated content, we are able to significantly improve the efficiency of the pre-training process and the stability of the optimization procedure.

## 5.2 Tokenization




(a) Minor variations in prompts such as a trailing space visualized by ␣ can drastically change the prediction.

| Ablation | Prompt | Exact Match |
|---|---|---|
| Sailor-1.8B | no space | 40.88 |
| | with space | 38.41 |
| *w.o.* BPE dropout | no space | 38.94 |
| | with space | 18.76 |

(b) Experiments on the TydiQA dataset indicate that applying BPE dropout significantly enhances the robustness of the Sailor-1.8B model when handling trailing spaces.

Figure 4: Initially, Sailor models were trained on 200B tokens using a greedy tokenization strategy. Subsequently, they were fine-tuned using BPE dropout for an additional 2B tokens, with a dropout rate of 0.1. As observed, BPE dropout improves the robustness.

**BPE Dropout for Robust Tokenization** We have observed that the model is unreasonably sensitive to small variations of the prompt, especially on *spaces*. As illustrated in Figure 4a, when prompting the model with the string "Answer:" without any trailing space yields a substantially improved performance compared to "Answer: ␣"[13]. The same phenomenon is observed in Qwen1.5, Mistral and Llama 2, and a similar issue has been discussed at lm-evaluation-harness library[14] (Gao et al., 2023). We attribute this kind of vulnerability to the tokenization strategy used in data processing. Modern tokenization methods usually employ the Byte Pair Encoding (BPE) (Sennrich et al., 2016) under the greedy segmentation setting[15], which means that sentences are segmented into subwords using the optimal tokenization strategy. The always-optimal strategy can make models vulnerable to unexpected subwords, such as an unexpected space in "Answer: ␣". To address this, we use BPE-Dropout during continual pre-training to randomly alter the BPE segmentation, providing subword regularization. While BPE-Dropout slightly increases loss on greedy subword segmentation, it improves both model performance and robustness, as demonstrated in Figure 4b.

**Vocabulary Expansion** We have tried our best to do vocabulary expansion on models like Mistral (Jiang et al., 2023a) and Llama-2 (Touvron et al., 2023b). However, similar to the observation in concurrent works (Zhao et al., 2024), it is challenging to expand the vocabulary with maintaining the original performance.

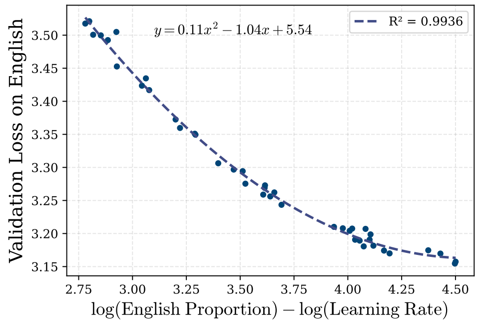
## 5.3 Training

In continual pre-training, we explore various configurations of learning rates and language data mixture. Starting with small proxy models, we randomly select learning rates from 20 intervals within a log range of 1e-5 to 4e-4, allowing efficient ex-

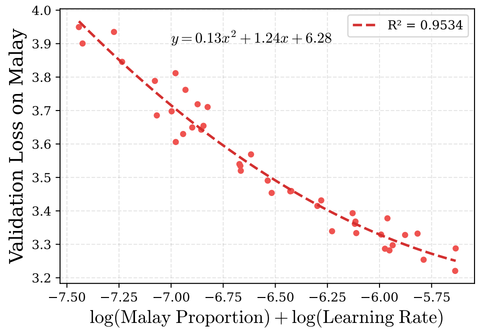
[13] We use "␣" to represent space.

[14] `https://github.com/EleutherAI/lm-evaluation-harness/issues/614`

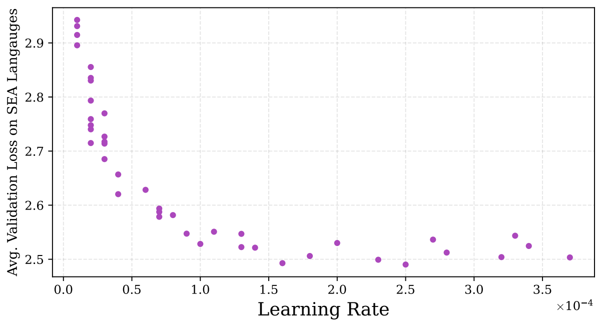
[15] The default BPE class is initialized with no dropout in the HuggingFace `tokenizers` library.




(a) The relationship between English loss and $\log(\text{English Proportion}) - \log(\text{Learning Rate})$.




(b) The relationship between Malay loss and $\log(\text{Malay Proportion}) + \log(\text{Learning Rate})$.




(c) The average SEA loss with increasing the learning rate.

Figure 5: Quadratic function between language proportion and learning rate.

perimentation. By evaluating English and SEA languages trade-offs on these models, we identify an optimal learning rate. We then fine-tune the data mixture to balance loss across languages, as detailed in Sec 2.1, for final model training.

**Learning Rate Tuning** The loss trend on the source domain (i.e., English) is primarily influenced by two factors: the proportion of English data during continual pre-training and the learning rate. Under the same token budget, the model's loss on English can be accurately modeled as a quadratic function of $\log(\text{English Proportion}) - \log(\text{Learning Rate})$, as shown in Figure 5a. In summary, increasing the learning rate, while holding the English data proportion constant, may negatively impact the model's performance on English.

Meanwhile, the loss trend on the target domain (i.e., SEA languages) is also mainly affected by the proportion of the target domain and the learning rate. However, there is a different modeling among the model loss on SEA languages, the proportion and the learning rate, as demonstrated by Figure 5b. From the observation, it becomes evident that the learning rate serves as a crucial hyper-parameter. A well-tuned learning rate plays a pivotal role in striking a balance between the acquisition of SEA languages and the forgetting of English. As shown in Figure 5c, considering that increasing the learning rate beyond 1e-4 does not yield significant improvements in the loss on SEA languages, we set the peak learning rate to 1e-4 in our experiments.

**Best Practise for Continual Pre-training** Drawing from the above insights, we highlight the importance of selecting the learning rate and the proportion of source domain data to mitigate catastrophic forgetting. We focus on the proposed quadratic function, which we refer to as the *magic metric* below. We suggest the following steps:

1. Fit a parametric quadratic function modeling the relationship between $\text{loss}_{\text{source}}$ and the magic metric via experiments varying learning rates and proportions.
2. Estimate the boundary of the magic metric value beyond which the model's $\text{loss}_{\text{source}}$ starts to deviate significantly from the original one.
3. Balance the learning progress on the target domain with the retention rate on the source domain by selecting a suitable magic metric larger than the boundary.
4. If the magic metric substantially exceeds the estimated boundary, it indicates that the model retains more knowledge from the source domain; conversely, it facilitates a more rapid learning pace on the target domain.

## 6 Conclusion

In this paper, we present the Sailor family of open language models (Apache License 2.0), tailored for South-East Asian languages, which exhibit strong performance across various multilingual tasks and benchmarks, fostering advancements in multilingual language models for the SEA region.

## Ethics Statement

All datasets and models used in this paper are publicly available, and our usage follows their licenses and terms. While we have made efforts to ensure safety and accuracy, our open-source language models may produce inaccurate, misleading, or potentially harmful content. Users must conduct their own safety assessments and implement necessary security measures before deployment. Usage must comply with local regulations. The authors bear no liability for any damages or claims arising from the use of these models, code, or demos.

## References


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.

AI Singapore. 2023. Sea-lion (southeast asian languages in one network): A family of large language models for southeast asia. `https://github.com/aisingapore/sealion`.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4623–4637. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. *CoRR*, abs/2308.16884.

Jonathan H. Clark, Jennimaria Palomaki, Vitaly Nikolaev, Eunsol Choi, Dan Garrette, Michael Collins, and Tom Kwiatkowski. 2020. Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Trans. Assoc. Comput. Linguistics*, 8:454–470.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. `https://github.com/open-compass/opencompass`.

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

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *Preprint*, arXiv:2310.01377.

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. *Preprint*, arXiv:2305.14233.

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.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. 2024. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *ArXiv*, abs/2310.06825.

Zixuan Jiang, Jiaqi Gu, Hanqing Zhu, and David Z. Pan. 2023b. Pre-rmsnorm and pre-crmsnorm transformers: Equivalent and efficient pre-ln transformers. *ArXiv*, abs/2305.14858.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. MADLAD-400: A multilingual and document-level large audited dataset. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Wing Lian, Guan Wang, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification.

Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. 2024. Regmix: Data mixture as regression for language model pre-training. *arXiv preprint arXiv:2407.01492*.

Quan Nguyen, Huy Pham, and Dung Dao. 2023a. Vinallama: Llama-based vietnamese foundation model. *CoRR*, abs/2312.11011.

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. 2023b. Seallms - large language models for southeast asia. *CoRR*, abs/2312.00738.

Kunat Pipatanakul, Phatrasek Jirabovonvisut, Potsawee Manakul, Sittipong Sripaisarnmongkol, Ruangsak Patomwong, Pathomporn Chokchainant, and Kasima Tharnpipitchai. 2023. Typhoon: Thai large language models. *CoRR*, abs/2312.13951.

Edoardo Maria Ponti, Goran Glavas, Olga Majewska, Qianchu Liu, Ivan Vulic, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2362–2376. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Ahad Rana. 2010. Common crawl – building an open web-scale crawl using hadoop.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *ArXiv*, abs/1909.08053.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Vu Minh Chien, Sebastian Ruder, Surya Guthikonda, Emad A. Alghamdi, Sebastian Gehrmann, Niklas Muennighoff, Max Bartolo, Julia Kreutzer, Ahmet Üstün, Marzieh Fadaee, and Sara Hooker. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. *Preprint*, arXiv:2402.06619.

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.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2022. Roformer: Enhanced transformer with rotary position embedding. *Preprint*, arXiv:2104.09864.

Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. 2024. Scaling laws with vocabulary: Larger models deserve larger vocabularies. *arXiv preprint arXiv:2407.13623*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

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, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, D. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, A. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. 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, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023c. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, Chenxia Li, Liu Yang, Xilin Luo, Xuejie Wu, Lunan Liu, Wenjun Cheng, Peng Cheng, Jianhao Zhang, Xiaoyu Zhang, Lei Lin, Xiaokun Wang, Yutuan Ma, Chuanhai Dong, Yanqi Sun, Yifu Chen, Yongyi Peng, Xiaojuan Liang, Shuicheng Yan, Han Fang, and Yahui Zhou. 2023. Skywork: A more open bilingual foundation model. *CoRR*, abs/2310.19341.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4003–4012. European Language Resources Association.

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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *CoRR*, abs/2401.02385.

Wenxuan Zhang, Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2023. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Jun Zhao, Zhihao Zhang, Luhui Gao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llama beyond english: An empirical study on language capability transfer. *CoRR*, abs/2401.01055.

| 3-shot (EM) | Thai | Indonesian | Vietnamese |
|---|---|---|---|
| Llama-2-7B | 31.78 | 39.78 | 38.00 |
| Mistral-7B-v0.1 | 34.33 | 41.33 | 41.33 |
| Typhoon-7B | 36.56 | – | – |
| VinaLLaMA-7B | – | – | 39.56 |
| Sea-Lion-7B | 36.33 | 35.56 | 37.00 |
| SeaLLM-7B-Hybrid | 37.78 | 43.11 | 43.00 |
| SeaLLM-7B-v2 | 36.33 | 43.11 | 47.00 |
| Qwen1.5-0.5B | 29.89 | 26.89 | 30.22 |
| Sailor-0.5B | 32.22 | 30.89 | 32.33 |
| Qwen1.5-1.8B | 30.11 | 32.00 | 31.33 |
| Sailor-1.8B | 34.22 | 34.89 | 35.33 |
| Qwen1.5-4B | 32.78 | 36.22 | 35.22 |
| Sailor-4B | 36.11 | 41.33 | 38.89 |
| Qwen1.5-7B | 38.33 | 42.00 | 42.89 |
| Sailor-7B | 41.56 | 44.33 | 45.33 |
| Qwen1.5-14B | 41.44 | 46.22 | 40.33 |
| Sailor-14B | 42.11 | 47.56 | 45.89 |

Table 4: Experimental results of different models on the Belebele benchmark.

## A Experimental Results

Detailed experimental results of different models on reading comprehension (Table 4), examination (Table 5), question answering (Table 6) and commonsense reasoning (Table 7) tasks.

| 3-shot (EM) | M3Exam (Thai) | M3Exam (Javanese) | M3Exam (Vietnamese) |
|---|---|---|---|
| Llama-2-7B | 21.13 | 23.99 | 34.15 |
| Mistral-7B-v0.1 | 29.59 | 31.00 | 43.54 |
| Typhoon-7B | 36.71 | – | – |
| VinaLLaMA-7B | – | – | 36.95 |
| Sea-Lion-7B | 23.90 | 21.56 | 26.89 |
| SeaLLM-7B-Hybrid | 25.98 | 24.53 | 38.79 |
| SeaLLM-7B-v2 | 35.60 | 29.92 | 50.36 |
| Qwen1.5-0.5B | 22.38 | 22.10 | 29.12 |
| Sailor-0.5B | 21.87 | 28.84 | 23.53 |
| Qwen1.5-1.8B | 23.81 | 26.15 | 36.39 |
| Sailor-1.8B | 23.90 | 29.65 | 27.67 |
| Qwen1.5-4B | 26.26 | 30.19 | 40.02 |
| Sailor-4B | 27.23 | 29.11 | 31.58 |
| Qwen1.5-7B | 35.88 | 33.15 | 51.09 |
| Sailor-7B | 38.33 | 35.85 | 51.98 |
| Qwen1.5-14B | 43.18 | 35.04 | 58.47 |
| Sailor-14B | 48.22 | 39.89 | 60.54 |

Table 5: Experimental results of different models on the examination task.

| 3-shot (EM / F1) | XQuAD (Thai) | TydiQA (Indonesian) | XQuAD (Vietnamese) |
|---|---|---|---|
| Llama-2-7B | 30.64 / 43.80 | 56.64 / 72.14 | 46.96 / 66.16 |
| Mistral-7B-v0.1 | 48.48 / 63.27 | 63.54 / 78.73 | 53.72 / 72.75 |
| Typhoon-7B | 51.70 / 68.92 | – | – |
| VinaLLaMA-7B | – | – | 44.82 / 64.81 |
| Sea-Lion-7B | 43.52 / 59.75 | 50.09 / 67.72 | 42.43 / 61.17 |
| SeaLLM-7B-Hybrid | 49.70 / 67.62 | 50.62 / 75.21 | 49.62 / 70.74 |
| SeaLLM-7B-v2 | 34.55 / 55.13 | 52.21 / 77.00 | 46.19 / 72.11 |
| Qwen1.5-0.5B | 14.19 / 23.35 | 20.71 / 32.64 | 19.85 / 35.38 |
| Sailor-0.5B | 15.84 / 27.58 | 30.44 / 54.74 | 21.13 / 40.57 |
| Qwen1.5-1.8B | 27.24 / 43.56 | 29.73 / 53.76 | 29.17 / 48.15 |
| Sailor-1.8B | 32.72 / 48.66 | 40.88 / 65.37 | 34.22 / 53.35 |
| Qwen1.5-4B | 34.03 / 53.40 | 48.32 / 72.68 | 43.71 / 63.86 |
| Sailor-4B | 46.82 / 63.34 | 53.98 / 73.48 | 47.65 / 67.09 |
| Qwen1.5-7B | 53.79 / 69.30 | 57.17 / 77.28 | 56.63 / 76.99 |
| Sailor-7B | 57.88 / 71.06 | 60.53 / 75.42 | 53.81 / 74.62 |
| Qwen1.5-14B | 55.53 / 74.36 | 60.18 / 81.05 | 57.57 / 77.58 |
| Sailor-14B | 49.43/ 69.99 | 58.94 / 77.85 | 57.83 / 77.37 |

Table 6: Experimental results of different models on the question answering task.

| 3-shot (EM) | XCOPA (Thai) | XCOPA (Indonesian) | XCOPA (Vietnamese) |
|---|---|---|---|
| Llama-2-7B | 52.80 | 64.00 | 62.00 |
| Mistral-7B-v0.1 | 57.20 | 62.40 | 61.60 |
| Typhoon-7B | 55.40 | – | – |
| VinaLLaMA-7B | – | – | 68.20 |
| Sea-Lion-7B | 60.80 | 60.60 | 67.80 |
| SeaLLM-7B-Hybrid | 58.20 | 71.60 | 67.60 |
| SeaLLM-7B-v2 | 56.80 | 64.00 | 64.60 |
| Qwen1.5-0.5B | 51.00 | 52.20 | 53.80 |
| Sailor-0.5B | 51.00 | 58.20 | 58.00 |
| Qwen1.5-1.8B | 52.60 | 51.60 | 53.40 |
| Sailor-1.8B | 53.80 | 64.20 | 63.20 |
| Qwen1.5-4B | 53.40 | 55.00 | 57.80 |
| Sailor-4B | 53.40 | 69.20 | 68.20 |
| Qwen1.5-7B | 54.20 | 62.20 | 66.20 |
| Sailor-7B | 59.00 | 72.20 | 72.20 |
| Qwen1.5-14B | 60.00 | 72.20 | 74.00 |
| Sailor-14B | 64.40 | 79.60 | 80.40 |

Table 7: Experimental results of different models on the commonsense reasoning task.