KORMo: Korean Open Reasoning Model for Everyone

Minjun Kim[1,4*] **Hyeonseok Lim**[1,4*] **Hangyeol Yoo**[1,4*] **Inho Won**[1*]
Seungwoo Song[1,4] **Minkyung Cho**[1] **Junghun Yuk**[1] **Changsu Choi**[1,4]
Dongjae Shin[1,4] **Huije Lee**[2] **Hoyun Song**[1]
Alice Oh[3] **KyungTae Lim**[1]

[1]KAIST MLP Lab [2]KAIST NLPCL Lab [3]KAIST U&I Lab [4]SeoulTech
Corresponding Author: ktlim@kaist.ac.kr

Abstract

This work presents the first large-scale investigation into constructing a *fully open* bilingual large language model (LLM) for a non-English language, specifically Korean, trained predominantly on synthetic data. We introduce **KORMo-10B**, a 10.8B-parameter model trained from scratch on a Korean–English corpus in which 68.74% of the Korean portion is synthetic. Through systematic experimentation, we demonstrate that synthetic data, when carefully curated with balanced linguistic coverage and diverse instruction styles, does not cause instability or degradation during large-scale pretraining. Furthermore, the model achieves performance comparable to that of contemporary open-weight multilingual baselines across a wide range of reasoning, knowledge, and instruction-following benchmarks. Our experiments reveal two key findings: (1) synthetic data can reliably sustain long-horizon pretraining without model collapse, and (2) bilingual instruction tuning enables near-native reasoning and discourse coherence in Korean. By fully releasing all components including data, code, training recipes, and logs, this work establishes a transparent framework for developing *synthetic data–driven fully open models (FOMs)* in low-resource settings and sets a reproducible precedent for future multilingual LLM research. All model checkpoints, datasets, and source codes are publicly available at `huggingface.co/kormo-lm`.

1 Introduction

Open-source large language models (LLMs) have recently demonstrated performance and utility approaching that of proprietary models, which has led to their rapid adoption in both academia and industry [Grattafiori et al., 2024, Team, 2025a, DeepSeek-AI et al., 2025a]. However, a significant number of these are *open-weight model (OWM)*, where only the final parameters are released, while critical components of the training recipe, such as data, preprocessing methods, code, hyperparameters, and training logs, often remain undisclosed. This limited scope of disclosure weakens the *chain of custody* necessary for reproducibility, fair comparison, and responsible deployment, ultimately hindering the scalability of subsequent research.

In response, *fully open models (FOMs)*, which transparently release the entire training pipeline, have emerged as a compelling alternative. The efficacy of the FOM approach was shown by OLMo et al. [2025], Muennighoff et al. [2024], who revealed their full methodology, including data sources, cleaning protocols, training scripts, hyperparameters, logs, and model checkpoints. While Workshop et al. [2023] stands as a prominent multilingual example, the trend towards full disclosure has also extended to explorations of compact, long-context, and reasoning-focused models [Bakouch et al.,

[*]marks core contributors.

Preprint. Under review.

2025a]. This trend has significant academic and societal impacts by enabling reproducible research and fostering an ecosystem for derivative models.

Despite these advancements, FOM development remains predominantly focused on English. In non-English settings, creating an FOM is made more difficult by several challenges: (i) a lack of large-scale web-crawled corpora, further complicated by copyright and licensing issues; (ii) the significant cost of data curation, including quality refinement, deduplication, and contamination control; and (iii) the ambiguities related to tokenizer architecture and language composition design. Nevertheless, it has been observed that releasing a foundational FOM for a specific language dramatically lowers research barriers and accelerates ecosystem development [Workshop et al., 2023, OLMo et al., 2025]. Therefore, demonstrating the successful implementation of a *first* non-English FOM is a task of profound academic and practical significance.

More recently, the use of synthetic and augmented data has emerged as a prominent method for addressing this disparity. Research by Gunasekar et al. [2023], Li et al. [2023] has demonstrated that textbook-style synthetic data contributes to enhanced efficiency in small- to medium-scale models. At the same time, large-scale synthesis pipelines are also being deployed in industrial applications [Nvidia et al., 2024]. Simultaneously, the data ecosystem is undergoing rapid enhancement through the public release of refined web corpora suitable for extensive, long-term pre-training [Soldaini et al., 2024, Su et al., 2025, Wang et al., 2025a, Penedo et al., 2024]. The application of synthetic data is prevalent not only in pre-training but also during the SFT and RLHF phases [Wang et al., 2023, Ouyang et al., 2022, Rafailov et al., 2023, Hong et al., 2024, Ethayarajh et al., 2024, Shao et al., 2024], with prompt engineering and data curricula being critical factors for model stability. However, the risk of model collapse due to the self-consuming nature of synthetic data has been a persistent concern [Shumailov et al., 2023a, Alemohammad et al., 2023]. Consequently, before synthetic data can be leveraged as a *primary resource* in non-English domains, rigorous quantitative validation of its stability and potential biases is imperative.

The design of the tokenizer and the language mixture ratio are also critical factors that determine the success or failure of a non-English FOM. The subword tokenization scheme (e.g., BPE, Unigram, byte-level), vocabulary size, and the proportion of each language directly impact compression efficiency, training cost, and downstream generalization [Sennrich et al., 2016, Kudo and Richardson, 2018, Radford et al., 2019, Chai et al., 2024]. This consideration is especially critical for non-Latin script and morphologically rich languages, where surface-form unit selection and vocabulary boundaries can operate differently. Therefore, in the context of an FOM, these aspects must be explicitly explored during the initial design phase.

We conduct a systematic investigation into the feasibility and limitations of constructing a *synthetic data-driven* FOM for Korean. We developed a Korean-English bilingual model *from scratch*, wherein synthetic data comprises 68.73% of the Korean corpus. We curated a training curriculum totaling 150 billion tokens. This curriculum was applied across all major stages of the training pipeline: pre-training, supervised fine-tuning (SFT), and preference learning. To achieve a varied distribution of styles and topics, the synthetic data was produced using a combination of Qwen and models from open-source families (e.g., GPT-OSS). This framework serves as the foundation for addressing the following research questions:

1. **RQ1. Stability**: Does synthetic data introduce long-term adverse effects on core components such as normalization, attention, and stabilization techniques? [Shumailov et al., 2023a, Alemohammad et al., 2023]
2. **RQ2. Tokenizer**: When using a high proportion of synthetic data, what is the optimal configuration of tokenizer, vocabulary size, and language mixture ratio for balancing compression efficiency against generalization? [Sennrich et al., 2016, Kudo and Richardson, 2018, Chai et al., 2024]
3. **RQ3. Bias**: When generating synthetic data, are the linguistic and cultural biases of the source model transferred to the new data, damaging or erasing the subtle nuances of the target language? [Wang et al., 2023, Ouyang et al., 2022]

Our research design consists of two stages. First, in a cost-effective *proxy* setting (1B model, 60B tokens), we directly compare a 100% synthetic data condition with a 100% non-synthetic condition. In this stage, we cross-analyze stabilization techniques such as RMSNorm/Pre-LN and curricula for learning rate, batch size, and sequence length, along with different tokenizers and language mixture

ratios. Once stability and efficiency are confirmed, then we proceed to build and release a 10.8B-scale Korean–English *fully open* model. We adhere to FOM principles by disclosing the entire pipeline, including data sources, filtering rules, scripts, hyperparameters, logs, and checkpoints [OLMo et al., 2025, Workshop et al., 2023]. The evaluation covers a wide range of standard benchmarks in knowledge, reasoning, reading comprehension, common sense, and mathematics (e.g., MMLU [Hendrycks et al., 2020b], MMLU-Pro [Wang et al., 2024b], ARC [Clark et al., 2018b], HellaSwag [Zellers et al., 2019a], etc.).

This paper presents the following contributions: (1) We are the first to systematically demonstrate that it is feasible to build an FOM in a non-English language, even with a *majority proportion of synthetic data*. (2) We examine how various tokenizer settings, language mixture ratios, and training curricula affect the trade-offs between *stability, efficiency, and generalization*, offering practical guidelines based on our results. and (3) By releasing a 10.8B parameter Korean-English *fully-open* model, including data, code, recipes, and logs, we significantly improve the reproducibility and accessibility of multilingual FOM research.

2 Exploring Training Design Choices

The training process of a large language model (LLM) involves numerous design decisions, spanning from the model architecture and scale (e.g., depth, width, number of layers), to hyperparameter settings like learning rate and its schedule, and strategies for the composition and preprocessing of training data. Previous research has reported the effectiveness of an approach where various configurations are first explored using smaller-scale *proxy* models, with the final model's architecture then being determined based on these initial results [Kaplan et al., 2020, Hoffmann et al., 2022]. This approach has proven to be a practical strategy for balancing cost-effectiveness with exploration speed. More recently, scaling priorities from a compute-optimal perspective are also being refined [Hoffmann et al., 2022].

While our study also follows this approach, we give special consideration to a scenario focused on the use of *augmented (synthetic) data*. Given that we use large-scale synthetic data as a primary resource, it is necessary to quantitatively verify the potential risks that this data might pose to model training, such as performance degradation from *self-consuming* loops [Shumailov et al., 2023a, Alemohammad et al., 2023]. We therefore trained a 1B-scale *proxy* model on 60B tokens to investigate two questions: (1) What are the performance differences between a model trained solely on synthetic data versus one trained on non-synthetic data? (2) Does synthetic data have a negative impact on training stability (e.g., the frequency of loss spikes)? Through this analysis, we experimentally validated the *effectiveness* and *stability* of a synthetic data-driven training approach.

In addition, we systematically evaluated core design factors in the tokenizer training process, including the composition of the data mixture, the tokenization compression ratio, and the influence of synthetic data on the overall distribution. The choice of tokenizer (BPE/Unigram/byte-level), vocabulary size, and language mixture ratio have been emphasized in numerous studies as factors that govern training efficiency and generalization performance. These are critical design considerations that must be examined, especially when compensating for data scarcity with synthetic data in non-English contexts.

We conducted a series of exploratory experiments as specified in Table 1. The synthesis of these results guided the final design of the KORMo(10.8B) model. The final design incorporates Pre-LN[Xiong et al., 2020] and RMSNorm [Zhang and Sennrich, 2019] for stable training; Grouped-Query Attention (GQA) [Ainslie et al., 2023] and Multi-Query Attention (MQA) [Shazeer, 2019] for inference efficiency; as well as SwiGLU activation [Shazeer, 2020] and RoPE positional embeddings [Su et al., 2024b]. Additionally, we utilized a document packing technique for large-scale corpus training, which efficiently fills sequences to their maximum length to reduce training waste [Chowdhery et al., 2023, Grattafiori et al., 2024]. This series of decisions provides practical design guidelines for building reproducible and open large-scale language models for non-English languages from scratch.

2.1 Experiment Settings

To explore the design choices for model architecture and training methods, a baseline training corpus, evaluation benchmarks, a *proxy* model, and a default training configuration are necessary. This study

Architecture Details	
Number of Total Parameters	1.33B
Number of Embedding Parameters	525M
Number of Non-Embedding Parameters	805M
Vocabulary Size	128256
Hidden Size	2048
Intermediate Size	6144
Number of Hidden Layers	16
Number of Attention Heads	16
Number of Key/Value Heads	8
Head Dimension	128
Attention Dropout	0.0
Attention Bias	0.0
Weight tying	False
Hidden Activation	`SwiGLU`
Normalizer	RMSNorm
RMS Norm Epsilon	$1e-05$
RoPE Theta	$5e+5$
Data type	`bfloat16`

Table 1: Proxy Model Default Configurations

configures the experiments as follows to make scale-up decisions based on patterns observable even in small-scale models (a *proxy-to-target* transfer). Given the ongoing debate about the *emergent* abilities reported in large-scale models [Wei et al., 2022, Schaeffer et al., 2023], we interpret the performance trends observed at the *proxy* stage with a focus on *relative comparison*.

1. **Language:** In the proxy stage, to ensure rich benchmark resources and experimental reliability, we limited our scope to English data to compare synthetic versus non-synthetic setups.

2. **Train Corpus:** For non-synthetic data, we used 60B tokens randomly sampled from *Ultra-FineWeb*, which has undergone multi-dimensional filtering [Wang et al., 2025a, Penedo et al., 2024]. During training, document *packing* was applied to match the maximum sequence length, thereby minimizing data waste [Chowdhery et al., 2023]. For the synthetic data experiments, we used 60B tokens from the *high-quality synthetic* split of *Nemotron-CC* [Su et al., 2025]. (While a broader comparison of various synthetic sources would be ideal, there were constraints in terms of data construction and training costs.)

3. **Evaluation Suite:** For the proxy model evaluation, we used MCQA benchmarks with balanced answer distributions and varied difficulty levels. General language understanding and reading comprehension were assessed with RACE [Lai et al., 2017], BoolQ [Clark et al., 2019a], and TruthfulQA (TFQA) [Lin et al., 2022]. Science and reasoning were evaluated with ARC-Easy (ARC-E) [Clark et al., 2018b] and OpenBookQA (OBQA) [Mihaylov et al., 2018]. Commonsense reasoning was evaluated using HellaSwag (HSWG) [Zellers et al., 2019a], Winogrande (WGRD) [Sakaguchi et al., 2021b], and PIQA [Bisk et al., 2020b]. For consistent comparison, a default 5-shot setting was used, but due to the nature of the task, TFQA was evaluated in a 0-shot setting (to prevent the model from fabricating confident answers).

4. **Model Architecture:** The base structure followed the Llama-3 series architecture, which includes *Pre-LN*, *GQA* [Ainslie et al., 2023], *SwiGLU* [Shazeer, 2020], and *RoPE* [Su et al., 2024b] [Grattafiori et al., 2024].

5. **Tokenizer:** Since the proxy experiments were conducted solely on English data, we used the well-established BPE-based tokenizer from Llama-3 [Grattafiori et al., 2024]. In Chapter 3, a tokenizer trained on English-Korean *bilingual* data is evaluated separately to reflect the actual target environment.

2.2 Normalization Methods

Normalization plays a key role in both training stability and performance improvement, and it is broadly categorized into *post-LN* and *Pre-LN* depending on its placement. Xiong et al. [2020] theoretically and empirically analyzed the differences in initialization and gradient behavior between the two methods, showing that *Pre-LN* converges more stably in large-scale pre-training. Additionally, *RMSNorm* was proposed as an alternative to reduce the computational cost of LayerNorm [Zhang and Sennrich, 2019], and stabilization techniques for very deep transformers (hundreds to thousands of layers), such as DeepNorm, have also been reported [Wang et al., 2024a]. More recently, hybrid approaches (such as *MixLN*) that mix different normalization methods in the initial and later layers have been discussed in this context, aiming to mitigate the gradient vanishing problem in very deep networks.

In this study, we hypothesized that the choice of normalization is directly related to **RQ1: Does augmented data negatively affect training stability?** This is because if augmented data exacerbates instability under a specific normalization method, it could become a critical risk factor in large-scale training.

Norm. type	data	arc_e	boolq	hswg	obqa	piqa	race	tfqa_mc1	tfqa_mc2	wgrd	AVG
Pre-LN	Web	65.24	56.54	37.86	23.0	70.24	31.77	19.95	32.84	52.96	43.38
MixLN	Web	60.52	52.54	35.18	20.6	68.23	29.19	20.44	36.13	48.70	41.28
Pre-LN	Synthetic	68.81	58.81	38.10	23.8	72.58	32.54	25.70	35.67	54.62	**45.63**

Table 2: Performance comparison across normalization methods (Pre-LN, MixLN) and data types. Pre-LN consistently outperforms MixLN, and the use of synthetic data introduces no performance degradation.

In the experiment, MixLN was configured by applying Post-LN to the first two layers, which constitute 12.5% of the total layers, and Pre-LN to the remainder. As shown in Table 2, Pre-LN was consistently superior to MixLN in average performance (43.38% vs. 41.28%). Therefore, Pre-LN was adopted as the default normalization method in the final architecture. A more critical observation is that when comparing the model trained with synthetic (augmented) data against the non-synthetic data model under the same Pre-LN setup, no performance degradation or increase in *loss spikes* was observed. This suggests that, from a normalization standpoint, synthetic data does not introduce additional instability, providing **positive evidence for RQ1.**

However, these results were obtained from a 1B parameter *proxy* model. Techniques like the MixLN family or stabilization methods for very deep networks such as DeepNorm may show more prominent potential at a larger model scale [Wang et al., 2024a]. Therefore, a re-evaluation is necessary for models with tens to hundreds of billions of parameters.

2.3 Attention Masking Method

Attention masking has a significant impact on both training efficiency and performance because it directly dictates how context is formed in long sequence *packing*. We compared four masking strategies using a *proxy* model, and improved implementation efficiency by leveraging flash-style kernels [Dao et al., 2022, Dao, 2023].

- **Causal masking:** The standard method in which every token attends to all previous tokens.
- **Sliding causal masking:** Tokens attend only to other tokens within a predefined window (1024 in this experiment). Windowed attention for long contexts is widely used in models like Longformer [Beltagy et al., 2020].
- **Intra-document causal masking (Intra-doc):** Blocks cross-document attention, allowing tokens to attend only to other tokens within the same document. A recent comparative study reports that *intra-doc* can improve downstream performance by reducing the noise that arises between documents during packing [Zhao et al., 2024].
- **Sliding intra-doc masking:** Combines intra-doc masking with a sliding window.

According to Table 3, Intra-doc masking achieved the highest performance with an average of 44.48%, surpassing standard causal masking (43.38%). This suggests that the strategy of strengthening the

Masking type	data	arc_e	boolq	hswg	obqa	piqa	race	tfqa_mc1	tfqa_mc2	wgrd	AVG
Causal	Web	65.24	56.54	37.86	23.0	70.24	31.77	19.95	32.84	52.96	43.38
Sliding causal	Web	67.13	54.43	37.75	23.0	71.22	32.54	19.22	32.61	52.09	43.33
Intra-doc	Web	67.68	56.88	38.10	23.8	71.71	31.87	20.20	35.67	54.38	44.48
Sliding Intra-doc	Web	67.80	53.33	37.92	22.4	70.89	31.96	20.69	34.64	52.72	43.59
Intra-doc	Synthetic	68.81	61.10	39.40	21.8	72.58	34.35	22.40	36.98	53.51	**45.66**

Table 3: Performance evaluation across different attention masking strategies and data types. Intra-doc masking was the most effective, and the model trained on synthetic data achieved higher performance under the same conditions.

independent context of each document by eliminating unnecessary contextual connections between them is effective [Zhao et al., 2024]. In contrast, the sliding-based methods may degrade performance by cutting off necessary long-range dependencies in long documents (similar observations are reported in the literature on windowed attention Beltagy et al., 2020).

Based on these results, Intra-doc masking was adopted for the final model. Under the same Intra-doc setting, the model trained on synthetic data performed better than its non-synthetic counterpart, with almost no difference in the learning curve and *loss* volatility. This serves as additional evidence that synthetic data does not impair stability from an attention masking perspective, which further supports our findings for RQ1. However, as this experiment was conducted in a monolingual English setting, the control strategy for document boundaries and cross-references needs further exploration in a multilingual context. For instance, we did not explore methods like Cross-Lingual Document Attention (XLDA, Han et al., 2025), which could be a promising direction for future work.

2.4 Multi-Token Prediction

Currently, large language models (LLMs) are primarily pre-trained and perform inference using the Next-Token Prediction (NTP) objective. However, the inherently sequential nature of NTP introduces significant latency, especially during inference, which constrains the real-time application of these models. [Gloeckle et al., 2024] Furthermore, its focus on predicting only a single token leaves room for optimization in terms of potential training efficiency. [Gloeckle et al., 2024] To overcome these limitations of NTP, Multi-Token Prediction (MTP) has emerged as a noteworthy methodology. [DeepSeek-AI et al., 2025b, Team, 2025b] MTP trains the model to predict several subsequent tokens simultaneously at each prediction step, offering two primary advantages: improved pre-training efficiency and accelerated inference speed.

In this context, we investigate the impact of the Multi-Token Prediction (MTP) objective, originally proposed in the DeepSeek-V3 architecture, on pre-training efficiency and downstream performance in small-scale language models.

Specifically, we set our primary goal to conduct an in-depth analysis of the applicability and effectiveness of MTP on a model with approximately 1 billion parameters. To integrate MTP into the base architecture, it is necessary to extend the conventional Next-Token Prediction (NTP; $n = 1$) loss to calculate a loss based on multi-token predictions.

Objective type	arc_e	boolq	hswg	obqa	piqa	race	tfqa_mc1	tfqa_mc2	wgrd	AVG
NTP	65.24	56.54	37.86	23.0	70.24	31.77	19.95	32.84	52.96	**43.38**
MTP	49.12	58.81	29.57	27.6	61.53	26.51	25.70	41.72	51.62	41.35

Table 4: Performance comparison of NTP and MTP training objectives. On the 1B-scale model, NTP showed higher overall average performance than MTP.

Table 4 presents a direct comparison between Next Token Prediction (NTP) and Multi-Token Prediction (MTP) under identical pre-training conditions. Overall, MTP showed slightly lower average performance compared to NTP (41.35 vs. 43.38, approx. –2.0%p), suggesting that NTP remains a more effective general-purpose training objective for small-scale models.

However, a different pattern was observed on a task-by-task basis. MTP outperformed NTP on Question Answering (QA) benchmarks such as BoolQ (+2.3%p), OBQA (+4.6%p), TFQA_mc1

(+5.8%p), and TFQA_mc2 (+8.9%p). This suggests that the training objective of predicting multiple tokens simultaneously provided a stronger signal for tasks requiring multi-step reasoning or factual recall. In contrast, NTP showed a clear advantage on tasks that rely on precise next-token prediction, such as ARC-E, PIQA, and RACE.

These results are consistent with findings reported in previous research Aynetdinov and Akbik [2025]. Small-scale language models may struggle to effectively handle the training complexity of MTP. Without sufficient capacity, they can show limitations in learning complex patterns and in generalization. In particular, the 1B parameter model used in this study, unlike models with tens of billions of parameters such as DeepSeek-V3 Mehra et al. [2025], can be interpreted as having structural and capacity constraints that prevent it from realizing the full potential of MTP.

The scale of the data is also a critical factor. This study used a 60B token English web corpus, which is significantly smaller than the trillions of tokens used by recent very large-scale models. For example, DeepSeek-V3 was trained on 14.8 trillion tokens, and it is likely that a complex objective like MTP is more effective with such vast amounts of data. In contrast, in a limited data environment, MTP might lead to overfitting or fail to sufficiently learn complex relationships as intended by its design.

In summary, we confirmed that NTP is a more stable and efficient training objective under the conditions of a small-scale model and limited data. Therefore, this study ultimately adopted NTP as the training objective, and no additional MTP experiments were conducted on the augmented data-based models.

3 Proposed Tokenizer based on Mixture of Datasets

A tokenizer serves as a fundamental component that transforms raw text into a sequence of tokens interpretable by language models. Achieving shorter sequences for identical text inputs, indicating superior compression, directly enhances both training and inference efficiency. Consequently, improving compression performance constitutes a central challenge in tokenizer design. In this study, we investigate strategies to enhance compression within bilingual environments and, from the perspective of RQ2, conduct an in-depth analysis of *the influence of synthetic data on tokenizer performance* [Chai et al., 2024].

3.1 Experiments Settings for Building Tokenizer from Scratch

The tokenizer employed in our experiments was trained using the *byte-level Byte Pair Encoding* algorithm, which has been widely adopted in recent large language models [Radford et al., 2019]. Since byte-level BPE begins segmentation at the byte level rather than the character level, it can process any Unicode text without information loss and effectively eliminates the out-of-vocabulary (OOV) problem.

Experimental Setup We analyzed the effect of the *synthetic data proportion* on compression when training tokenizers using the byte-level BPE algorithm. As synthetic data sources, we sampled 20 GB each from Cosmopedia (English) and ko-Cosmopedia (Korean, internally constructed) [Ben Allal et al., 2024]. As representative non-synthetic corpora, we sampled 20 GB each from UltraFineWeb (English) and FineWeb2 (Korean) [Wang et al., 2025a, Penedo et al., 2024]. For evaluation, a subset of The Pile was used for non-synthetic measurements, while a portion of RLHF datasets for each language was used for synthetic measurements [Gao et al., 2020, Bai et al., 2022]. These datasets were selected because, beyond general web domains, they encompass diverse fields, allowing a more comprehensive assessment of compression performance. Table 5 summarizes the data composition. The model follows the previously proposed configuration.

Evaluation Metrics Compression was measured using *Bytes Per Token* (BPT). A tokenizer that produces shorter token sequences for the same text yields a higher BPT, which indicates more efficient compression. However, since prior studies have shown that higher compression does not necessarily guarantee better downstream performance [Ali et al., 2024, Zuo et al., 2025], this study jointly evaluates both BPT and model downstream performance. To reflect the bilingual scenario, we evaluated downstream model performance using benchmark datasets in both Korean (Haerae, KMMLU, and KoBEST) [Son et al., 2023a, 2025a, Jang et al., 2022b] and English.

Split / Corpus		Lang	Category	Sampling	Size
Train	Cosmopedia	EN	Synthetic (encyclopedic)	Random	20 GB
	ko-Cosmopedia	KO	Synthetic (encyclopedic)	Random	20 GB
	UltraFineWeb	EN	Web (non-synthetic)	Random	20 GB
	FineWeb-2	KO	Web (non-synthetic)	Random	20 GB
Val. (Web mix)	The Pile (subset)	EN	Web mix	Random	33 MB
	culturaX (subset)	KO	Web mix	Random	33 MB
Val. (Synthetic)	RLHF (per-lang)	EN	Synthetic (RLHF)	Random	33 MB
	RLHF (per-lang)	KO	Synthetic (RLHF)	Random	33 MB

Table 5: Tokenizer compression study setup. We train a byte-level BPE tokenizer and analyze how *synthetic* corpora (Cosmopedia, ko-Cosmopedia) versus *non-synthetic web* corpora (UltraFineWeb, FineWeb-2) affect compression performance. Each training corpus is randomly sampled to 20 GB. Validation uses (i) web-crawled subsets (*The Pile*, *cultura-ko-x*) and (ii) per-language synthetic RLHF data.

3.2 Impact of Synthetic Data on Tokenizer Training

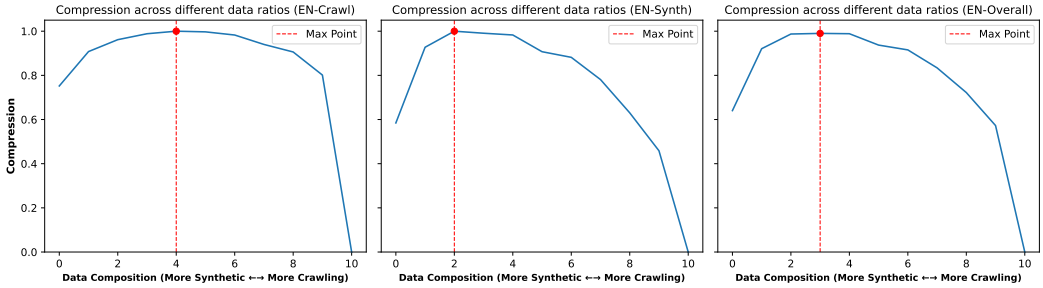


Figure 1: Compression trends by data ratio in the English setting. The x-axis represents the synthetic–crawl ratio (left: synthetic-dominant, right: crawl-dominant), and the y-axis shows compression efficiency measured in bytes per token (BPT), where higher values indicate greater efficiency.

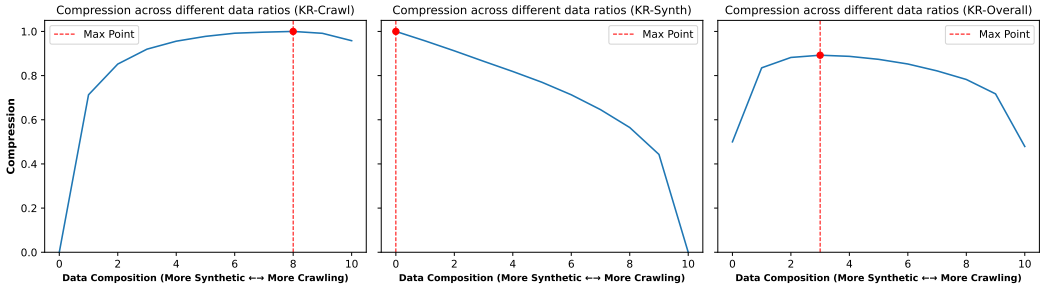


Figure 2: Compression trends by data ratio in the Korean setting. The x-axis represents the synthetic–crawl ratio (left: synthetic-dominant, right: crawl-dominant), and the y-axis shows compression efficiency measured in bytes per token (BPT), where higher values indicate greater efficiency.

Figure 1 and 2 present the results for English and Korean settings, respectively. The X-axis represents the ratio between synthetic and crawled data, while the Y-axis indicates compression performance measured in BPT. Overall, both languages exhibit an average compression gain as the proportion of synthetic data increases. A closer examination by domain, however, reveals several notable differences.

For English, even within crawled domains, higher compression was observed when the synthetic proportion reached approximately 60% (see Ali et al., 2024 for discussion on the potential trade-off between compression and generalization). In contrast, for Korean, a substantially higher crawled data proportion—around 80%—was required to achieve comparable compression. This suggests that the

distributional gap between synthetic and crawled data is more pronounced in Korean than in English, potentially reflecting differences in language-specific knowledge and character-level properties [Son et al., 2023a, 2025a, Jang et al., 2022b].

Alias	English		Korean		Code	Data Size	Vocab Size
	Crawling	Synthetic	Crawling	Synthetic			
EK-Crawl	25.5%	59.5%	4.5%	10.5%	-	20GB	125k 196k
EK	-	85%	-	15%	-	20GB	125k 196k
EPK	-	80%	-	5%	15%	20GB	125k 196k

Table 6: Data composition and configuration details of the proposed tokenizer candidates, showing the proportion of crawled, synthetic, and code data used for training under two vocabulary scales (125k and 196k).

Based on these observations, we constructed three tokenizer design candidates following three principles: (1) reflecting the optimal language-specific mixture, (2) leveraging the overall superiority of synthetic data, and (3) compensating for domain weaknesses. First, **EK-Ratio** incorporates the optimal synthetic–crawled ratios for each language, as observed in Figures 1 and 2, directly into the tokenizer training mixture. Second, **EK** maintains the same English–Korean ratio but trains exclusively on synthetic data for both languages to exploit its overall advantage. Third, **EPK** augments the mixture with additional code data to address weaknesses in code-related domains. Each tokenizer candidate was independently trained with vocabulary sizes of 125K and 196K. Subsequent analyses compare their BPT and downstream performance against commercial tokenizers.

3.3 Compression Comparison in Bilingual Settings with Commercial Tokenizers

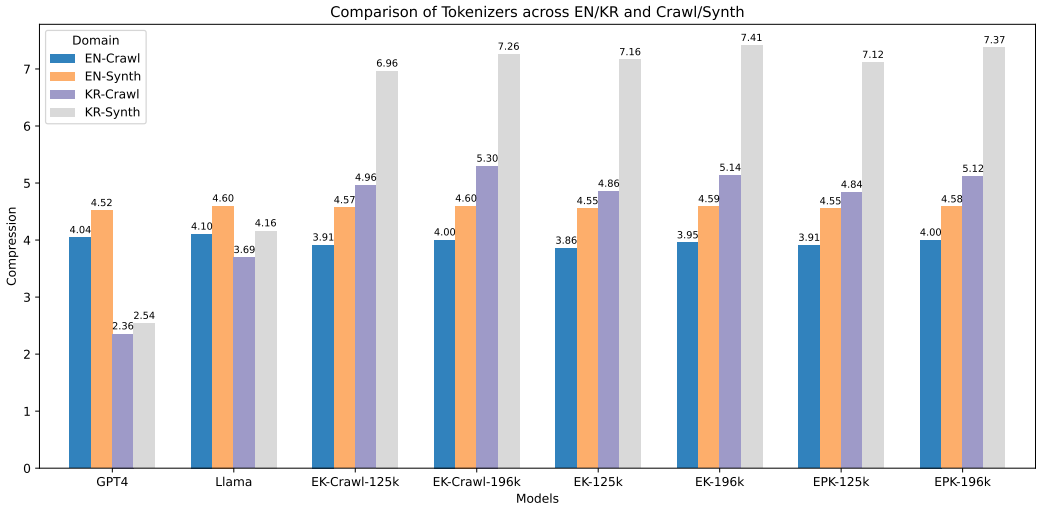


Figure 3: Comparison of English and Korean compression performance between the tokenizer candidates defined in Table 6 and commercial tokenizers (GPT-4, LLaMA).

We first examined whether the proposed tokenizer candidates achieve compression performance comparable to that of commercial models (e.g., LLaMA, GPT-4). Figure 3 summarizes the compression results for both our candidates and the commercial tokenizers. As expected, byte-level BPE–based tokenizers from the LLaMA family exhibit strong performance in English [Grattafiori et al., 2024, Radford et al., 2019], but their effectiveness in Korean is relatively limited. In contrast, the proposed tokenizer candidates maintained performance on par with commercial models in English domains,

while demonstrating a substantial advantage in Korean. This finding suggests that *bilingual* mixture and domain adjustment can enhance compression efficiency for non-Latin scripts [Ali et al., 2024, Chai et al., 2024]. We also examined the effect of vocabulary size on compression. Under identical training configurations, expanding the vocabulary generally improved compression by approximately 2–7%. This implies that while a larger vocabulary can yield compression benefits, it also introduces trade-offs related to model parameters and memory usage[Kudo and Richardson, 2018, Ali et al., 2024]. As noted earlier, compression is merely one of the intrinsic metrics for tokenizer quality. Thus, downstream model performance must also be evaluated [Ali et al., 2024]. Accordingly, we trained models using each tokenizer on 9B Korean tokens (FineWeb2) and 51B English tokens (Ultra-FineWeb) and assessed their performance across a range of downstream tasks. Since tokenization granularity differs across tokenizers, token counts were normalized using a single reference tokenizer (GPT-4 tokenizer) to ensure comparability at the "60B-token" scale [Radford et al., 2019].

3.4 Downstream Model Performance with Diverse Tokenizers

Model	arc_e	boolq	hswg	obqa	piqa	race	tfqa_mc1	tfqa_mc2	wgrd	AVG
LlamaTok	66.79	51.56	37.27	**24.80**	69.48	31.58	20.69	35.65	50.75	43.17
GPT Tok	67.34	47.98	37.53	23.80	70.35	30.53	19.46	35.66	53.75	42.93
Our EK 125k Tok	67.63	**58.93**	**38.42**	22.80	71.06	**32.92**	21.54	36.49	**54.62**	**44.93**
Our EK 125k with Crawl Tok	67.30	48.99	37.94	22.20	70.84	31.67	21.18	**38.99**	51.85	43.44
Our EPK 125k Tok	**67.68**	57.06	37.75	24.40	**71.49**	31.29	**23.26**	37.36	53.91	44.91
Our EK 196k Tok	67.55	48.38	37.32	23.20	70.95	31.10	20.81	38.71	52.64	43.41
Our EPK 196k Tok	68.39	46.57	38.00	24.40	70.89	32.54	21.30	38.23	52.88	43.69
Our EK 196k with Crawl Tok	67.97	55.87	37.89	25.40	70.18	31.87	22.28	36.90	52.64	44.56

Table 7: Comparison of English downstream model performance across different tokenizers.

Model	csatqa	haerae	kmmlu	KoBEST					AVG
				boolq	copa	hellaswag	sentineg	wic	
Llama Tok	17.65	20.17	13.80	**50.71**	54.90	**35.00**	51.39	51.03	36.86
GPT Tok	16.04	19.25	27.24	48.93	53.50	33.00	54.16	**51.91**	38.03
Our EK 125k Tok	24.06	**21.08**	24.24	49.72	53.60	31.00	55.92	50.64	38.83
Our EK 125k with Crawl Tok	19.79	18.42	27.38	49.29	**56.40**	34.00	56.93	49.52	38.97
Our EPK 125k Tok	**24.60**	20.17	**28.46**	50.21	53.50	31.00	**57.18**	49.84	**39.42**
Our EK 196k Tok	16.58	18.24	19.57	49.86	55.60	30.00	56.42	51.11	37.22
Our EPK 196k Tok	19.79	21.72	21.94	51.43	56.60	33.00	47.10	50.87	37.81
Our EK 196k with Crawl Tok	17.65	19.80	24.91	50.50	56.30	34.00	56.42	50.79	38.67

Table 8: Comparison of Korean downstream model performance across different tokenizers. All evaluations were conducted under a 3-shot setting.

The downstream performance results for both English and Korean are summarized as follows. In English, **EK 125K** achieved the best overall performance, followed by **EPK 125K**. Notably, despite commercial tokenizers exhibiting superior compression in English, models trained with our proposed tokenizers consistently achieved higher downstream performance. This finding suggests that the relationship between compression efficiency and downstream generalization is non-monotonic [Ali et al., 2024], highlighting that tokenizer design should be jointly optimized with the data mixture and the model's downstream performance rather than treating them as independent factors.

In Korean, the proposed tokenizers also ranked among the top performers, with **EPK 125K** demonstrating the highest results.

Meanwhile, vocabulary expansion yielded consistent gains in compression but tended to reduce average downstream performance in English (with a few exceptions such as EK-with-Crawl). Based on these observations, we excluded the **196K**-vocabulary tokenizers from the final set of candidates [Ali et al., 2024].

3.5 Safety of KORMo Tokenizers

This section examines the presence of potentially harmful tokens (e.g., toxic or biased expressions) within the vocabularies of the proposed tokenizer candidates and compares them with Korean-specialized commercial models (Exaone, HyperCLOVA X, A.X, and Midm). The harmfulness

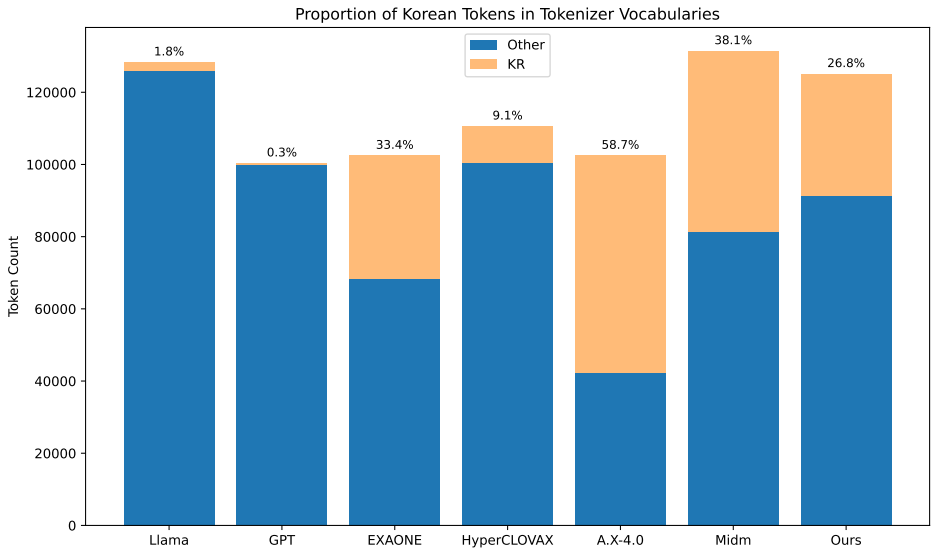


Figure 4: Proportion of Korean tokens within tokenizer vocabularies across different models. Each bar represents the share of Korean (*KR*) versus non-Korean (*Other*) tokens. While English-centric models such as LLaMA and GPT exhibit minimal Korean coverage (1.8% and 0.3%, respectively), Korean-specialized models (Exaone4, HyperCLOVAX, A.X-4.0, Midm and ours) show significantly higher proportions.

Tokenizer	**Trained Token with Bias**
Llama	출장안마, 마사지, 출장마사지, 콜걸
EXAONE	새끼, 마사지, 시발, 몰카, 새끼야, 씨발, 개새끼, 야한, 지랄, 카지노, 시발, 토토, 젖꼭지, 좆같, 자위, 섹스, 좆, 추천인
HyperCLOVAX-SEED	놀이터토토, 역출장샵추천, 출장만남, 동출장만남, 역출장, 역출장만남, 동출장맛사지후기, 사설놀이터, 토토사이트, 동출장샵, 토토, 보지, 안마후기, 출장맛사지, 출장마사지, 동출장, 출장샵추천, 출장대행, 동출장안마, 바카라, 먹튀, 동출장대행, 마사지, 카지노, 동출장마사지, 동콜걸출장마사지, 동콜걸추천, 면출장대행, 면출장만남, 면출장마사지, 영화무료보기어플, 역출장마사지, 면출장샵추천, 역출장대행, 출장안마, 동출장아가씨, 콜걸추천, 콜걸출장마사지, 동출장샵추천, 동출장만남후기
A.X-4.0	토토, 자위, 좆, 병신, 카지노, 애미, 자지, 섹스, 시발, 새끼
Midm	모텔, 토토, 보지, 자위, 좆, 몰카, 카지노, 애미, 자지, 섹스, 시발, 새끼
EK-Crawl	안출장안마, 출장미인아가씨, 예약금없는출장샵, 홍출장안마, 토토사이트, 온라인카지노, 출장업계, 토토, 양출장안마, 성출장안마, 출장샵, 코인카지노, 먹튀, 출장걸, 콜걸업소, 출장업계위, 출장만남, 지역출장마사지샵, 호텔카지노, 출장, 더킹카지노, 출장소이스홍성, 출장만족보장, 출장가격, 출장코스가격, 출장업소, 출장샵예약, 출장만족, 콜걸샵, 카지노사이트, 우리카지노, 에티오피, 출장마사지, 출장최고, 캐츠비카지노, 모텔출장, 출장부르는법, 타이마사지, 출장외국인, 출장샵안내, 출장연애인급, 콜걸후기, 출장샵콜걸, 콜걸출장마사지, 카지노하는곳, 역출장안마, 솔레어카지노, 미시출장안마, 출장서비스, 출장소, 출장최장미녀, 바카라사이트, 보지, 시출장샵, 콜걸출장안마, 출장소이스, 콜걸, 예스카지노, 콜걸강추, 바카라하는곳, 콜걸만남, 카지노, 모텔출장마사지샵, 출장샵강추, 마사지황형, 출장샵추천, 출장최고시, 주출장안마, 출장서비스보장, 츠비카지노, 콜걸추천, 외국인출장만남, 릉콜걸샵, 출장샵예약포항, 출장아가씨, 바카라, 출장색시미녀언니, 출장오쓰피걸, 출장몸매최고, 오피걸, 레어카지노, 마사지, 천출장안마, 킹카지노, 출장여대생, 출장샵후기, 동출장마사지, 새끼, 출장오피, 산출장안마, 전지역출장마사지샵, 출장마사지샵, 출장안마, 출장전화번호, 출장맛사지
EK	보지, 카지노, 젖, 새끼
EPK	보지, 카지노, 젖, 새끼

Table 9: Examples of biased or potentially harmful tokens identified in Korean-specialized tokenizers.

evaluation was conducted based on established toxicity and bias benchmarks such as RealToxicityPrompts, HateXplain, and HateCheck [Gehman et al., 2020, Mathew et al., 2021, Röttger et al., 2021].

Table 9 summarizes the harmful or potentially sensitive tokens identified in each tokenizer's vocabulary. We found that several Korean-specialized models also contained a non-negligible number

of harmful tokens, with the HyperCLOVA X family exhibiting relatively higher proportions. In contrast, English-centric models such as LLaMA and GPT showed minimal Korean coverage in their vocabularies under our counting criteria, thereby limiting the detection of harmful Korean tokens—an observation that itself suggests constraints in their expressive capacity for Korean. Figure 4 further confirms the very low proportion of Korean tokens in these models.

Overall, the proposed candidates (EK and EPK) contained comparatively fewer harmful tokens. However, the inclusion of *crawled* data tended to increase the number of such tokens. This finding indicates that data source characteristics can influence the formation of bias during tokenizer training. In light of this concern, we excluded crawled data from the final tokenizer training configuration [Penedo et al., 2024, Wang et al., 2025a]. Considering all the aforementioned factors, **EPK-125K** was selected as the final tokenizer.

4 Pretraining Phase

This section introduces our proposed approach for collecting and generating pre-training data, as well as the methodology for constructing training stages that account for data quality and difficulty.

4.1 Pretraining Datasets

Language	Dataset Name	# tokens	# origin tok	Reasoning	Synthetic	Synthesizer	Seed	Stage
English	DCLM[2]	1,000B	6,000B	X	X	-	-	stage 1
	UltraFineWeb[3]	793B	1,000B	X	X	-	-	stage 2
	Nemotron-CC-web[4]	280B	4,400B	X	X	-	-	stage 2
	Nemotron-CC-synthetic	1,000B	1,900B	X	O	Mistral-Nemo-12B-Instruct	Nemotron-CC-HQ	stage 2
	stack-edu[5]	152B	–	X	X	-	-	stage 2
	Fine-math[6]	37.3B	–	X	X	-	-	stage 2
	Cosmopedia[7]	25B	–	X	O	Mixtral-8x7B-Instruct-v0.1	Cosmopedia seed suite[8]	stage 2
	OpenCodeReasoning[9]	5.46B	–	O	O	DeepSeek-R1	-	stage 2
	OpenMathReasoning[10]	24.89B	–	O	O	DeepSeek-R1, QwQ-32B	-	stage 2
Korean	Ko-Web Datasets	36.3B	837B	X	X	-	-	stage 1
	Ko-CC-Dump	6.2B	251B	X	X	-	-	stage 1
	Korean Opensource	5.57B	–	X	X	-	-	stage 2
	Synth-FineWeb2	10.97B	–	X	O	Qwen3-30B-A3B	FineWeb2[11]	stage 2
	Synth-Nemo-HQ	32.82B	–	X	O	Qwen3-30B-A3B	Nemotron-HQ	stage 2
	Kosmopedia	4.07B	–	X	O	GPT-oss (120B)	cosmopedia	stage 2
	Synth-Ultrafineweb	41.69B	–	X	O	GPT-oss (120B)	Ultrafineweb	stage 2
	Ko-Reasoning	7.05B	–	O	O	Qwen3-235B-A22B	Nemotron-Post	stage 2
English + Korean total pretraining tokens: 3,462.32B								

Table 10: Comparison of English and Korean pretraining datasets used in KORMo. Each dataset is categorized by reasoning trace inclusion, synthetic nature, synthesizer model, and training stage. The underlined datasets indicate generated synthetic data produced in-house.

Pre-training is the most resource-intensive and labor-demanding stage in building a language model. For KORMo's pre-training, we collected and generated large-scale, high-quality text corpora. Table 10 summarizes the composition of the primary pre-training datasets. The data were obtained through both direct collection (web crawling and the use of publicly available datasets) and synthetic generation, encompassing general web text, domain-specific materials, and various auxiliary datasets.

4.1.1 Public Data Collection

English Public Data. We first collected publicly available high-quality datasets and utilized them for KORMo's pre-training. As summarized in Table 10, all English data were sourced from open

[2] github.com/mlfoundations/dclm

[3] huggingface.co/datasets/openbmb/Ultra-FineWeb

[4] data.commoncrawl.org/contrib/Nemotron/Nemotron-CC/index.html

[5] huggingface.co/datasets/HuggingFaceTB/stack-edu

[6] huggingface.co/datasets/HuggingFaceTB/finemath

[7] huggingface.co/datasets/HuggingFaceTB/finemath

[8] Web-text, stanford.edu, UltraChat, OpenHermes2.5, OpenStax, KhanAcademy, AutoMathText

[9] huggingface.co/datasets/nvidia/OpenCodeReasoning

[10] huggingface.co/datasets/nvidia/OpenMathReasoning

[11] huggingface.co/datasets/HuggingFaceFW/fineweb-2

datasets and further refined through additional filtering processes before use. Specifically, the web corpora provided by DCLM [Li et al., 2024], UltraFineWeb [Wang et al., 2025b], and Nemotron-CC [Su et al., 2024a] are high-quality datasets constructed through multi-stage filtering based on Common Crawl, serving as the foundation for general language understanding. In addition, Nemotron-synthetic is a synthetic dataset generated from Nemotron-web, while Cosmopedia is a synthetic corpus built using RefinedWeb [Penedo et al., 2023] and RedPajama [Weber et al., 2024] as seed data, included to enhance the generalization performance of large-scale language models. Additionally, for foundational training in mathematics and coding, we utilized the Stack-Edu and Fine-Math datasets, while the OpenCodeReasoning and OpenMathReasoning datasets were incorporated to further enhance higher-order reasoning abilities. Through this multi-layered data composition, KORMo was designed to acquire diverse forms of knowledge across multiple domains.

Korean Public Data. For Korean data, publicly available pre-training resources are extremely limited. Therefore, we collected data from two primary sources: existing open resources and raw dumps directly parsed from Common Crawl.

- **Korean Opensource:** We collected writing, news, and document summarization data from the National Institute of Korean Language's Modu Corpus[12], and included the academic paper dataset provided by KISTI[13]. These corpora are released by reputable Korean research institutions, ensuring both reliability and quality.
- **Ko-Web Datasets:** We utilized Korean datasets released by international research communities, including community-OSCAR[14], CulturaX[15], and FineWeb2 v2.0.0[16]. However, these datasets share a limitation in that their knowledge cutoff extends only up to 2023.
- **Ko-CC-Dump:** To ensure data freshness, we directly extracted Korean text from the raw dumps of Common Crawl. Specifically, we parsed WARC files from four dumps, 2025-13, 2025-18, 2025-20, and 2025-21, corresponding to web crawl data collected between March 15 and May 25, 2025. This approach follows the findings reported in DCLM, which demonstrated that directly parsing WARC files yields higher-quality data than relying on preprocessed WET files. For data extraction, we used the `datatrove` library and employed `resiliparse` to ensure efficient and stable processing.

  The language identification provided by Common Crawl (based on CLD2) exhibited low accuracy; therefore, we implemented a new language filtering process using fastText's LID-176 model. The classification threshold was set to 0.8 to minimize noise while maintaining high recall for Korean text. Although only four subsets were utilized in this study due to time constraints, all additional subsets will be released in the future to support further research[17].

4.1.2 Synthetic Data Generation

For non-English languages, particularly Korean, publicly available pretraining data with permissive licenses remains extremely scarce. This poses a fundamental limitation for building fully open-source LLMs. To address this challenge, we do not simply aim to increase the volume of training data. Instead, we empirically investigate methods for effectively transferring knowledge from English-centric corpora into Korean through controlled augmentation strategies.

Recent work, such as Nemotron-CC-HQ, has demonstrated the practical utility of augmenting limited seed data through various transformations, including QA-style conversions, sentence rephrasings, and Wikipedia-style rewrites. Building on this approach, we generate large-scale Korean data using high-quality English seed corpora as input. However, we note that directly transferring Anglocentric cultural contexts or lifestyle content may limit the downstream applicability of Korean models. To mitigate this, we introduced prompt design constraints that reflect Korean-specific context and usage scenarios. We also constructed multiple seed pools to promote diversity in format, narrative style, and content, even when using the same English input. In total, we constructed five Korean augmentation datasets, each derived from a distinct combination of English seed data and prompt strategy.

[12] `https://kli.korean.go.kr/corpus/main/requestMain.do?lang=ko`

[13] `https://aida.kisti.re.kr/data/`

[14] `https://huggingface.co/datasets/oscar-corpus/community-oscar`

[15] `https://huggingface.co/datasets/uonlp/CulturaX`

[16] `https://huggingface.co/datasets/HuggingFaceFW/fineweb-2`

[17] `https://huggingface.co/datasets/MLP-KTLim/Kor-CC-Resili-Parsed`

- **Synth-FineWeb2:** Korean responses generated from FineWeb2 seed data using custom-designed prompts.
- **Synth-UltraFineWeb:** Korean outputs generated by appending the instruction "answer in Korean" to Nemotron-CC English prompts.
- **Synth-Nemo-HQ:** Korean data generated by translating Nemotron-CC-HQ prompts into Korean prior to synthesis.
- **Kosmopedia:** Korean data generated from Cosmopedia seed texts using custom-designed Korean prompts.
- **Ko-Reasoning:** Constructed by pairing each English query from Nemotron-Post with a corresponding Korean reasoning trace and answer.

To ensure diversity, we generated augmented datasets from five distinct English sources. These sources differ in domain, narrative style, and knowledge density, enabling broad and high-quality knowledge transfer into Korean. However, as noted in RQ1, the use of synthetic data raises concerns about its potential impacts on training stability and long-term model performance. To assess this, we adopted an evaluation methodology similar to that proposed in the Nemotron-HQ study. Specifically, we applied our tokenizer to the proxy model selected in Section 2 and conducted pretraining with 4B tokens from each augmented dataset. We then evaluated the resulting models on downstream tasks. Experimental results show that the Kosmopedia-based data yielded the strongest performance, and most other augmented datasets also outperformed models trained on standard Korean web corpora. These findings provide empirical evidence supporting the robustness and effectiveness of synthetic data, as raised in RQ1, and suggest that large-scale synthetic pretraining is a viable strategy for building open-source LLMs in low-resource languages.

4.1.3 Data Filtering

Model performance is highly sensitive to the quality of the pre-training data, which is not only the largest in volume but also predominantly sourced from the web. To address this, various filtering techniques are applied to extract high-quality text from the raw collected data. In our work, we apply a three-stage filtering pipeline: Heuristic Filtering, Deduplication, and Quality Filtering. The specifics of this process are detailed below.

Heuristic Filtering We applied heuristic filtering to all collected Korean web data. Heuristic filtering refers to a rule-based approach for removing various forms of noise present in text. We identify and eliminate texts of extreme length, often caused by crawling errors, and instances of excessive repetition of special characters. Since such data be detrimental to model training, we strictly removed them through a well-defined pipeline. Our heuristic filtering pipeline was primarily based on the method proposed by [Li et al., 2024].

1. **Initial Preprocessing** – We first performed text normalization to ensure consistent processing across all samples. Consecutive spaces or tabs were reduced to a single space, sequences of three or more newline characters were replaced with two newlines, and any CR/CRLF sequences exceeding two were also normalized to two newlines. Samples that became empty after normalization were discarded.
2. **Word-Count Filter** – To remove documents with extreme lengths, which typically result from crawling errors, we applied a word-count filter. For computational efficiency, words were defined as space-delimited tokens. Documents with fewer than 10 or more than 10,000 words were discarded.
3. **Non-Alphabetic Word Ratio Filter** – We filtered documents containing a high proportion of non-alphabetic words (i.e., tokens composed entirely of digits or symbols). Samples with a ratio of such non-alphabetic words exceeding 0.25 were discarded.
4. **Alphabetic–Numeric Character Ratio Filter** – Conversely, samples in which the proportion of meaningful characters like letters or digits, within the entire text was too low were removed. Texts dominated by special symbols are likely to represent advertisements or structured markup rather than natural language. The cutoff threshold was set to 0.25.
5. **Symbol Ratio Filter** – Samples containing an excessive number of specific symbols relative to their word count were identified as noise and removed. This filter primarily targets template-like, spam, or table-corrupted text. The monitored symbols were

["#", "...", ". . .", "\u2026"], and samples exceeding a ratio of 0.1 were dropped.

6. **N-gram Repetition Filter** – Samples exhibiting excessive repetition of contiguous n-grams were removed, as they can induce undesirable repetition patterns during model training. We analyzed contiguous n-grams of lengths 8 to 10 and discarded any document with a repetition ratio exceeding 0.2.
7. **Line-Ellipsis Ratio Filter** – We removed documents where a large fraction of lines terminated with an ellipsis ({'...', '. . .', '\u2026'}). Such incomplete sentences are prevalent in web data and provide low-quality signals for language modeling. Documents exceeding a line-ellipsis ratio of 0.3 were dropped.
8. **Bullet Point Ratio Filter** – Documents where a high proportion of lines began with bullet symbols (['●', '•', '*', '-']) were removed, as they typically represent lists or navigational content rather than natural language prose. The removal threshold for this ratio was set to 0.9.

Removing duplicate or repetitive data is a crucial step in all stages of model training. To achieve this, we adopted the Bloom Filter–based deduplication method (BFF)[18] proposed in DCLM, following the original study's hyperparameter settings. The BFF framework provides two modes of deduplication: *document* and *old-both*. The *document* mode preserves more tokens but applies a relatively lenient deduplication threshold, whereas *old-both* enforces a stricter criterion for removing redundant samples.

we performed cross-deduplication sequentially across *FineWeb2*, *CulturaX*, our internally crawled *CC-Dump*, and *Community-OSCAR*. However, unlike for English, the volume of available training data in Korean is significantly more limited, making the decisions regarding the intensity of the filtering process particularly critical.

The *Deduplication* column in Table 11 reports the performance of Korean models trained using the Document and Old-both deduplication methods in combination with Quality Filtering (QF). As clearly shown in the results, models trained on data processed with the stricter Old-both method consistently outperform those trained with the Document method across all evaluation metrics. This suggests that in pre-training dataset construction, enhancing data quality by minimizing redundancy has a more decisive impact on final model performance than simply increasing the quantity of data.

However, applying the Old-both method resulted in the removal of approximately 70% of the entire Korean corpus, as those samples were identified as duplicates. This substantial reduction reveals that a large portion of publicly available open-source Korean datasets significantly overlap with one another, leaving a limited amount of truly unique and usable content. Despite the reduction, we selected the Old-both–deduplicated dataset as our final version, given its clear and consistent performance gains across all evaluation metrics.

Deduplication	Q·F	csatqa	haerae	kmmlu	kobest_boolq	kobest_copa	kobest_hellaswag	kobest_sentineg	kobest_wic	MMLU_proX_ko	AVG
Document	Ver1	18.717	19.157	26.389	50.000	61.1	46.6	63.476	51.508	10.911	37.740
Document	Ver2	25.134	18.698	25.826	50.997	59.9	47.8	71.788	50.873	10.834	**39.006**
Document	Ver3	19.786	18.973	27.436	52.493	62.5	49.2	50.882	50.635	10.707	37.001
Document	Ver4	16.578	18.882	20.645	51.282	59.3	49.2	60.705	50.079	10.418	36.188
Old-both	Ver1	19.251	19.890	26.100	53.276	64.0	50.2	58.186	50.556	10.860	37.947
Old-both	Ver2	17.647	18.882	28.030	52.279	64.5	49.2	77.330	50.952	10.664	**39.809**
Old-both	Ver3	16.578	19.157	21.259	51.282	64.1	51.8	62.972	51.032	10.613	37.399
Old-both	Ver4	17.112	17.415	27.976	53.632	65.5	51.4	53.904	52.302	10.375	37.313

Table 11: Korean benchmark performance(Accuracy) under different deduplication and quality-filtering strategies. 'Deduplication' refers to the performance difference based on the deduplication method supported by BFF[19], and 'Q·F' indicates the effect of quality filtering. All experiments use a 1B proxy model trained on 40B Korean tokens extracted from the full Korean corpus.

Quality Filtering Following cross-corpus deduplication to mitigate redundancy, we performed a quality filtering step to ensure the reliability of the training corpus. In large-scale language model pretraining, data quality is a pivotal factor influencing downstream performance. Web-sourced corpora often contain toxic content—including profanity, hate speech, and social biases—as well as low-information text such as spam, templated phrases, and boilerplate. Accordingly, rigorous filtering is

[18] https://github.com/mlfoundations/dclm/tree/main/dedup/bff

[19] https://github.com/allenai/bff

Samples	Category	Version 1	Version 2	Version 3	Version 4
Positive Samples (300k)	Korean-Opensource	122,340	122,340	90,000	90,000
	Instruction-Following	50,000	50,000	50,000	50,000
	Qwen-annotated (>3)	37,660	37,660	37,660	37,660
	Synthetic-LLM	90,000	90,000	122,340	122,340
Negative Samples (300k)	Qwen-annotated (=0)	✓		✓	
	Random crawl		✓		✓

Table 12: Comparison of dataset composition across Version 1–4. Versions 1 and 2 share identical positive samples but differ in negative sample selection, and likewise for Versions 3 and 4. Versions [1,2] emphasize balanced sources, while Versions [3,4] increase the proportion of synthetic datasets.

essential to exclude such low-quality samples and construct a cleaner, high-quality dataset suitable for robust model training.

Quality filtering is typically performed by assigning a score between 0 and 5 to each input text using a classifier and removing data below a set threshold. Both DCLM and UltraFineWeb [Wang et al., 2025b] report that fastText-based classifiers are efficient and effective for this task. For English web data, we additionally applied the fastText-based UltraFineWeb filter, despite the data having already undergone initial filtering, based on reports that it outperforms the DCLM-style classifiers used in earlier stages.

For Korean data, we constructed a dedicated quality classifier, as no pre-existing model was available. Following the approaches of DCLM and UltraFineWeb, we designed a fastText-based classifier tailored to Korean. Using translated scoring prompts from FineWeb-Edu [Penedo et al., 2024], we employed Qwen3-32B to assign scores (0–5) to approximately 400K samples from the Korean portion of community-OSCAR (Figure 5).

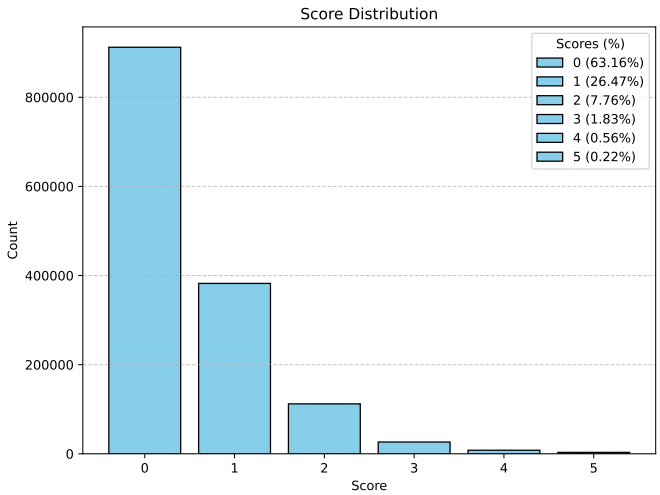


Figure 5: Distribution of quality scores assigned to 400K Korean community-OSCAR samples using Qwen3-32B with translated FineWeb-Edu scoring prompts. The majority of samples were rated as 0, indicating low or no educational value.

To account for variations in classifier effectiveness depending on the choice of positive and negative samples, we designed four versions of fastText-based quality classifiers (Table 12). Positive samples included: (i) Korean-Opensource (public datasets released by official institutions), (ii) Ko-Reasoning data, (iii) Qwen-annotated samples with scores ≥ 3, and (iv) Synth-FineWeb2 data. Negative samples consisted of: (i) Qwen-annotated samples with a score of 0, and (ii) randomly selected samples from community-OSCAR. Each classifier was trained on 300K positive and 300K negative samples. We

then applied each classifier to filter 40B tokens of Korean data and trained a KORMo-1B proxy model to assess filtering effectiveness (Table 11).

Our analysis yields two key insights:

1. **Effect of Negative Sample Composition:** Version 2, which utilized randomly sampled web data as negative examples, achieved higher average performance than Version 1, which used explicitly labeled low-quality (score 0) samples. This suggests that exposing the classifier to a broader spectrum of naturally occurring low-quality patterns on the web is more beneficial for generalization than training solely on narrowly defined "poor-quality" data.
2. **Effect of Positive Sample Composition:** Versions 1 and 2, primarily composed of curated, high-quality web data, consistently outperformed Versions 3 and 4, which contained a higher proportion of synthetic data. This result reaffirms the intrinsic value of high-quality, authentic data. However, due to the absolute scarcity of high-quality Korean web data, incorporating synthetic data remains necessary to acquire the trillions of tokens required for large-scale pre-training.

Overall, Version 2 proved to be the most effective filtering strategy. Accordingly, we adopted the Version 2–based fastText classifier to refine the entire Korean corpus used for KORMo's pre-training dataset.

4.2 Pretraining

We initiated pre-training only after completing all stages of data preparation. This section explores the core components of KORMo's pre-training from the following three perspectives:

- **Optimal Learning Rate Search:** Determining an appropriate learning rate configuration for the proposed architecture and data environment.
- **Language Ratio:** Adjusting the proportion of Korean and English data across training stages.
- **Synthetic Data Ratio:**Examining the potential limitations arising from an overreliance on synthetic data.

4.2.1 Optimal Learning Rate Search

One of the most critical early decisions in pre-training is the selection of an appropriate learning rate. The learning rate significantly influences both convergence speed and final performance: excessively high values can lead to divergence or unstable oscillations, while excessively low values may result in slow learning or convergence to a suboptimal state. Since the optimal learning rate can vary substantially depending on model size and architecture, we conducted a direct search using the target model, KORMo-10B.

KORMo-10B was built upon the proposed training design, incorporating Pre-LN architecture, intra-document masking, a next-token prediction objective, and a custom-developed bilingual tokenizer. The learning rate search was conducted with a global batch size of 1024 and a sequence length of 4096 over 2,000 steps, exploring candidate values in the range of {1e-4, 3e-4, 5e-4, 7e-4, 9e-4, 1e-3, 1.5e-3, 3e-3}. To closely approximate real training conditions, a randomly sampled subset of the entire pretraining corpus was used during this search.

The results presented in Figure 6 reveal distinct optimization patterns across different learning rates.

- High learning rates (e.g., 3e-3) led to rapid initial loss reduction but soon caused oscillations and instability.
- Low learning rates (e.g., 1e-4) exhibited highly stable convergence but progressed too slowly to reach optimal performance.
- Moderate learning rates (7e-4, 9e-4) achieved both stability and fast convergence, with 7e-4 yielding the lowest overall loss across all steps.

Accordingly, we adopted a learning rate of 7e-4 as the optimal setting for the final model training. This result demonstrates that, under the architecture and data environment of KORMo-10B, this configuration enables stable convergence and efficient optimization.

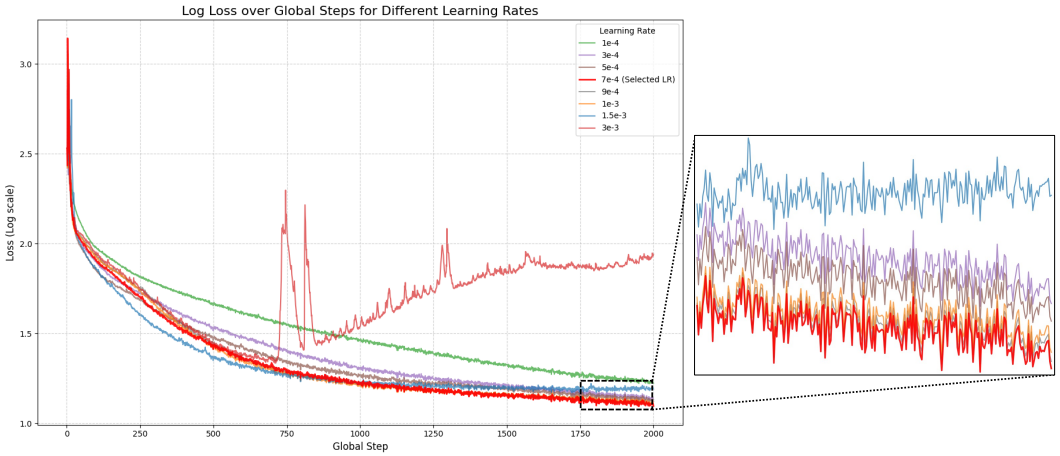
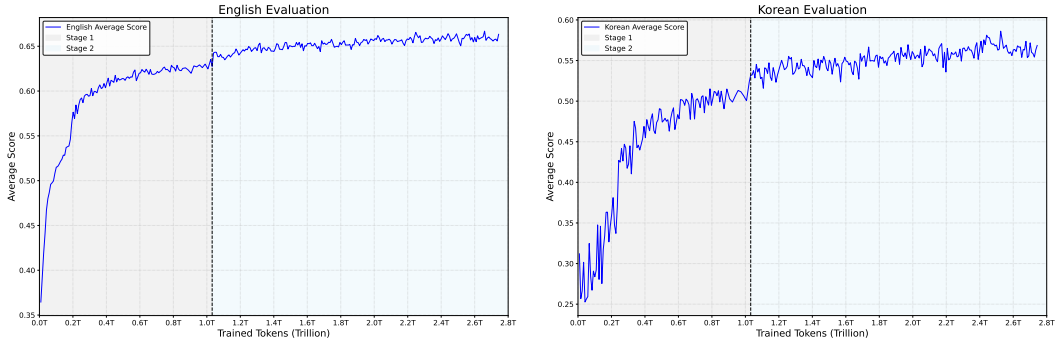


Figure 6: Comparison of loss curves across different learning rates during KORMo-10B pre-training.



(a) English Tasks Average Score

(b) Korean Tasks Average Score

Figure 7: Comparison of Average Performance between English and Korean Evaluation Tasks During Pretraining stages.

4.2.2 Proposed Language Ratio and Pre-training Stages

Language Proportion. In bilingual language models, the data ratio between the two languages serves as a crucial determinant of model performance. Previous studies have shown that even when the target language accounts for as little as 1.14% of the total data in a multilingual setting, a reasonable level of performance can still be achieved Xue et al. [2021]. In contrast, LLaMA-2 includes only about 0.06% Korean data, making it practically infeasible to assess Korean performance meaningfully. Although systematic investigations of language ratios in bilingual settings are limited, recent research has reported that increasing the target language proportion to 1.5–5% can still maintain competitive performance Seto et al. [2025]. However, such results may vary substantially depending on interlingual similarity and data quality. Taking into account the structural heterogeneity between English and Korean, this study set the Korean data proportion to over 5% to ensure stable bilingual learning.

Training Phase. Recent trends in LLM pre-training adopt multi-stage training strategies that progressively increase data quality to leverage the benefits of curriculum learning. In the case of KORMo, due to practical constraints that limit the total number of training tokens to below 3T under the chosen Korean–English ratio, we adopted a two-stage pre-training strategy.

- **Stage 1:** Approximately 1T tokens were trained primarily on relatively low-quality web data. Specifically, this stage included DCLM (960B), Korean Web (36.3B), and Korean-CC (6.2B) datasets corresponding to the “stage1” entries in Table 10. The objective of this stage

is to enable the model to acquire fundamental language understanding from large-scale web text and to develop robustness against noisy data.

- **Stage 2:** The model is trained on high-quality web text, synthetic data, and low-difficulty reasoning datasets that include short reasoning paths. As shown in the "stage2" entries of Table 10, this stage comprises a total of 1.8T tokens (1.7T in English and 0.1T in Korean). Building upon the foundational language abilities acquired in Stage 1, this phase aims to strengthen advanced language understanding and reasoning capabilities by incorporating structured and higher-quality data.

Architecture Details	
Number of Total Parameters	10.75B
Number of Embedding Parameters	1.02B
Number of Non-Embedding Parameters	9.73B
Vocabulary Size	125,184
Hidden Size	4096
Intermediate Size	16,384
Number of Hidden Layers	40
Number of Attention Heads	32
Number of Key/Value Heads	8
Head Dimension	128
Attention Dropout	0.0
Attention Bias	0.0
Weight tying	False
Hidden Activation	`SwiGLU`
Normalizer	RMSNorm
RMS Norm Epsilon	$1e-05$
RoPE Theta	$5e+5$
Data type	`bfloat16`

Table 13: KORMo-10B Configurations

Training Details. Table 13 summarizes the configuration of the KORMo-10B model. The key training settings are as follows.

- **Model Architecture:** Reflecting the findings presented in section 2, the model employs a transformer decoder with a Pre-LN architecture. Training is conducted using intra-document attention masking and a next-token prediction objective, while the flash-attention-3 kernel [Shah et al., 2024] is applied to improve computational efficiency.
- **Context length:** By default, sequence packing was applied based on a sequence length of 4096. However, for reasoning datasets, packing was avoided since it could negatively affect coherence by truncating connected reasoning chains. Instead, sample-level padding was applied without additional packing.
- **Weight decay:** Following the approach of the OLMo study [OLMo et al., 2024], weight decay was not applied to token embeddings. This decision was made to prevent undesirable side effects, such as excessive shrinkage of embedding vector magnitudes.
- **Initialize optimizer:** At the beginning of each stage, the optimizer is reinitialized to ensure stable adaptation to the shift in data distribution, and a learning rate warm-up of 0.03% of the total training steps is applied.
- **Training Resources:** All experiments were conducted in a multinode environment using NVIDIA H200 GPUs, with a total of 128 GPUs utilized in parallel. Considering the model size and number of parameters, the Fully Sharded Data Parallel (FSDP) [Zhao et al., 2023] strategy was adopted to enable parameter sharding across nodes and efficient memory utilization.

Figure 7 illustrates the average English and Korean evaluation performance as a function of the number of training tokens during pre-training. The English evaluation comprises 12 benchmarks—MMLU,

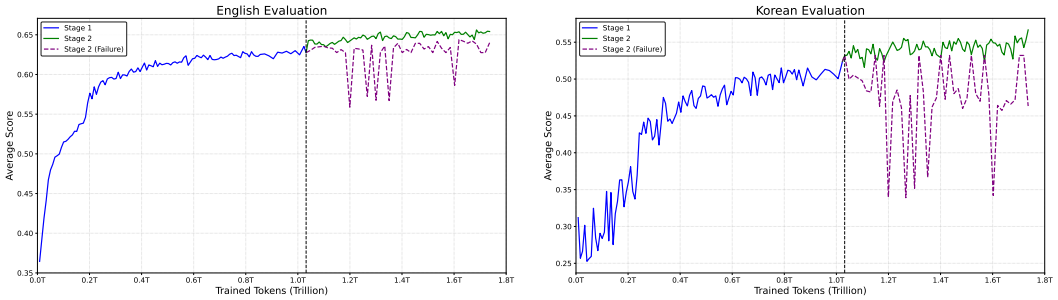
BoolQ, COPA, ARC-Challenge, ARC-Easy, AGIEval-EN, CommonsenseQA, OpenBookQA, PIQA, HellaSwag, SocialIQA, and Winogrande—and the Korean evaluation consists of six benchmarks: KMMLU, CSATQA, CLICK, Haerae, K2-Eval, and KoBEST.

The observations indicate that in Stage 1, both languages exhibited a sharp performance improvement, suggesting that the model rapidly acquired fundamental linguistic knowledge and syntactic structures. In Stage 2, the performance curve transitioned into a more gradual upward trend, implying that the high-quality data refined the model's existing capabilities and progressively enhanced its reasoning ability.

However, as shown in Figure 7b, the Korean performance curve exhibits more pronounced oscillations compared to the English curve in Figure 7a. This variability in Korean performance can be attributed to two main factors: (1) the relatively low proportion of Korean data in the pretraining corpus, which may lead to instability in internal representations; and (2) the smaller number of evaluation benchmarks for Korean (six) compared to English (twelve), causing individual benchmark fluctuations to have a greater impact on the overall average. Nevertheless, by the end of Stage 2, Korean performance stabilized at an average score above 0.55, confirming that the KORMo model achieved robust bilingual performance.

Meanwhile, when the total number of training tokens exceeded 2.8T, additional performance gains were observed. However, considering cost-effectiveness, we concluded that allocating the same computational resources to mid-training or post-training stages would be a more reasonable strategy.

4.3 Impact of Augmented Data Diversity



(a) English Tasks Average Score

(b) Korean Tasks Average Score

Figure 8: Graph illustrating a failed pretraining case using a single synthetic dataset.

KORMo began leveraging large-scale augmented data in Stage 2. However, we observed that insufficient diversity in the augmented data at this stage led to severe adverse effects during model training. The Stage 2 (Failure) curve in Figure 8 presents results obtained under the same configuration as the standard Stage 2 setup, except that the Korean augmented data consisted solely of samples generated by a single synthesizer, Synth-Nemo-HQ. This comparison allows us to examine the impact of using augmented data generated from multiple synthesizers versus that produced by a single synthesizer on language model training.

Examining the performance of the Stage 2 (Failure) model reveals that neither language maintained the peak performance achieved in Stage 1; instead, both experienced overall degradation accompanied by pronounced oscillations. Specifically, English performance fluctuated unstably within the range of 0.55–0.63, while Korean performance declined even more sharply, reverting to levels comparable to the early Stage 1 phase. This abrupt and irreversible degradation is interpreted as a manifestation of model collapse.

The primary causes of this collapse can be inferred as follows:

1. **Single-Model-Based Synthetic Data:** All Korean data were generated solely from a single model (Qwen3-30B-A3B), which may have caused the repeated learning of that model's inherent biases and errors Shumailov et al. [2024], Long et al. [2024], Bukharin et al. [2024]. Particularly in refined training stages such as Stage 2, the lack of diversity in synthetic data

led to a rapid loss of the model's broad knowledge distribution Shumailov et al. [2023b], Chen et al. [2024], Gerstgrasser et al. [2024].

2. **Uniformity of Seed and Prompt Types:** When the seed data and prompt types are limited, the resulting synthetic data distribution becomes narrow. This constraint can lead the model to overfit to restricted patterns, potentially causing the loss of general language understanding and reasoning abilities acquired during Stage 1 Bukharin et al. [2024].

In summary, while the use of augmented data is essential for high-quality training, our findings demonstrate that relying on limited seeds, models, or prompt types in synthetic data generation can lead to severe performance degradation. Therefore, in large-scale bilingual language model development, ensuring not only the quantity but also the diversity of augmented data is imperative for stable and effective learning.

5 Mid-training

Recent language models have diversified according to their intended applications, leading to the gradual standardization of the *mid-training* stage. In reasoning-oriented models, the ability to comprehend long contexts and internalize complex reasoning processes is essential, thereby necessitating an intermediate training phase following pre-training [Liu et al., 2024, Gao et al., 2025a]. The KORMo model was similarly designed with two primary objectives for its mid-training stage: (1) to extend context length and (2) to enhance reasoning capability.

5.1 Long Context Training

Recent studies have shown that reasoning tasks require long-form outputs containing complex reasoning traces. Consequently, long-context training during the mid-training phase is essential to ensure that models can reliably process both long inputs and outputs [Liu et al., 2024]. Following recent studies, we adopted a similar long-context training procedure for KORMo to enhance its capability in complex reasoning and long-text comprehension.

Data Preparation. For the English dataset, we followed the approach proposed in ProLong [Gao et al., 2025b], utilizing the `prolong-data-64k` corpus while truncating sequences to a maximum length of 32k tokens. For the Korean dataset, we reconstructed large-scale long-form texts into book-level documents and repacked them to approximately 32k tokens, thereby adapting them for long-context training.

In addition, prior studies have reported that training only on long documents may degrade performance in domains such as mathematics and code [Dong et al., 2025]. To mitigate this issue, we adopted a mixed training strategy that incorporates relatively short, high-quality data. Specifically, short Korean synthetic datasets were also packed to roughly 32k tokens and integrated into the long-context training corpus to balance the model's ability between long-context comprehension and short-form reasoning [Gao et al., 2025a].

Language	Dataset Name	Origin	# tokens
English	ProLong	princeton-nlp/prolong-data-64K	7.21B
Korean	Ko-book	AI-Hub	1.11B
	Kosmopedia	cosmopedia (English)	0.51B
	Koneedle	Ko-Web Datasets (in Table 10)	1.44B
English + Korean total Reasoning tokens: 10.27B			

Table 14: Overview of Korean and English datasets used in the Reasoning Mid Train stage, including their sources and token counts.

Furthermore, during Korean long-form QA tasks, we observed that the model produced responses in English due to limited Korean exposure. To address this issue, we constructed Needle-in-a-Haystack-style data, designed to evaluate and improve the model's ability to retrieve specific information

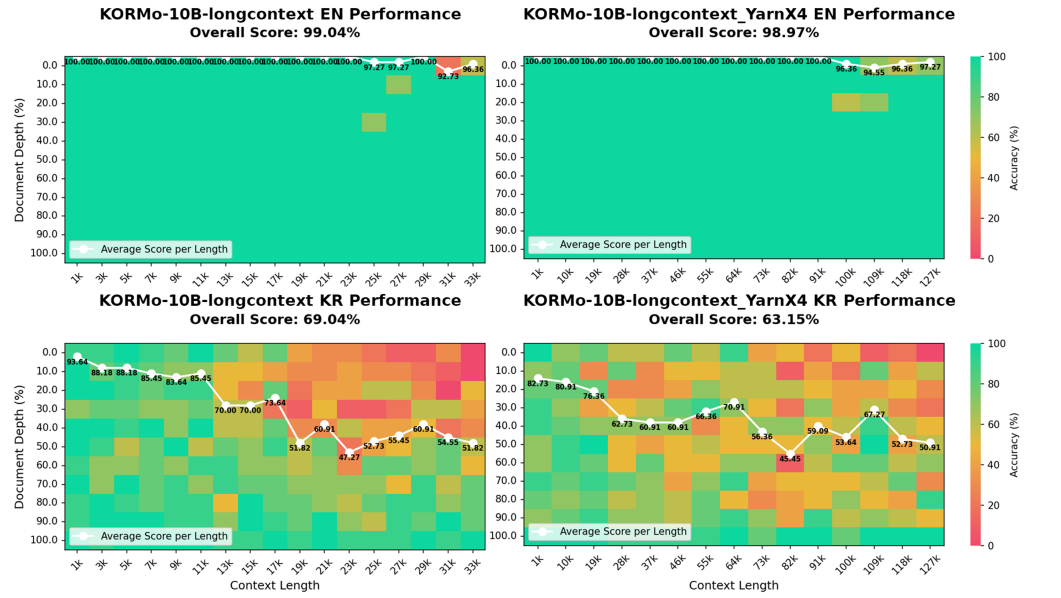


Figure 9: Performance of KORMo-10B models on the Needle In A Haystack benchmark for English (top row) and Korean (bottom row). Color intensity denotes retrieval accuracy per document depth and context length.

randomly inserted into 32k-token documents. The final datasets used for long-context extension consisted of 7.21B tokens in English and 3.06B tokens in Korean.

Experimental Results on Long Context Extension To evaluate KORMo's ability to retrieve information within long contexts, we employed a English–Korean bilingual version of the Needle In A Haystack (NIAH) benchmark. This benchmark measures a model's capacity to accurately locate and extract specific information randomly inserted within documents of up to 127K tokens in length.

Figure 9 presents the evaluation results. The baseline KORMo-10B-longcontext model achieved an accuracy of 99.04% in English and 69.04% in Korean, while the YarnX4-extended model recorded 98.97% in English and 63.15% in Korean.

For English, both models maintained almost perfect performance across all input lengths, demonstrating stable retrieval capability even beyond the 32K context window. This result confirms the robustness and effectiveness of the long-context training procedure.

In contrast, Korean performance gradually declined beyond the 13K token range, dropping below 60% accuracy after 21K tokens. Although the YarnX4-based model exhibited greater stability in certain segments, a degradation appeared between 82K and 118K tokens.

These results suggest that for Korean, stable information retrieval capabilities in long contexts have not yet been fully established. This outcome may be attributed to a complex interplay of factors, including tokenizer instability, imbalances in the synthetic data, and domain bias within the long-context dataset. Consequently, further improvements in data design and augmentation strategies specifically tailored for Korean long-context learning are required.

5.2 Reasoning Context Training

The second phase of mid-training, *Reasoning Context Training*, aims to improve the model's reasoning ability by training on datasets containing large-scale reasoning traces. This stage is performed after long-context extension and is designed to efficiently elicit reasoning capabilities and help the model internalize advanced inference patterns Bakouch et al. [2025b], Team [2025b].

Data Preparation. Table 5.2 summarizes the dataset composition used in this stage. Prior studies have shown that emphasizing reasoning traces in STEM, coding, and math domains can improve

general model performance Bakouch et al. [2025b], DeepSeek-AI et al. [2025b], Team [2025b]. Based on this observation, we prioritized STEM, coding, and math domains when constructing the subset from Nemotron-Post-v1, and incorporated the full OpenThoughts dataset to broaden coverage. To further support general reasoning ability, we added QA pairs from the reasoning-relevant portion of Nemotron's chat domain.

We reuse the Ko-Reasoning corpus constructed during pretraining (Section 4.1), as there are no publicly available large-scale reasoning datasets for Korean. Since Ko-Reasoning contains limited coverage of STEM, coding, and math domains, we supplement it by translating selected portions of the Nemotron dataset. To ensure effective reasoning supervision, we apply difficulty-based filtering and retain only high-quality samples that demonstrably require reasoning traces.

Algorithm 1 The two-stage filtering algorithm for selecting high-difficulty reasoning seeds for translate. Stage 1 selects for samples incorrectly answered by two models, and Stage 2 filters for a consensus on high difficulty.

Input: Annotated dataset D with model answers and difficulty labels
Output: Final high-difficulty subset D'
$P \leftarrow \emptyset$ ▷ candidate pool
$D' \leftarrow \emptyset$ ▷ final dataset for translation

Stage 1: Non-Reasoning Sample Filtering
for $d \in D$ **do**
  if IsIncorrect(Qwen-30B) **and** IsIncorrect(Qwen-4B) **then**
    $P \leftarrow P \cup \{d\}$ ▷ Add only samples where both models failed
  end if
end for

Stage 2: Difficulty Consensus
for $d \in P$ **do**
  if IsHighDifficulty(Qwen-30B) **and** IsHighDifficulty(Qwen-4B) **then**
    $D' \leftarrow D' \cup \{d\}$ ▷ Add only samples where both models agree on high difficulty
  end if
end for

Return: D'

Language	Dataset Name	Synthesizer	# tokens	# num rows
English	Nemotron-Post-Training-Dataset-v1[20]	Qwen3-235B & DeepSeek-R1	144.75B	24.9M
	OpenThoughts3-1.2M[21]	QwQ-32B	5.46B	0.4M
Korean	Ko-Reasoning 10	Qwen3-235B	7.05B	5.8M
	Nemotron-Post-Trans	GPT-oss(120B) & Qwen3-Next-80B	2.83B	1.1M
English + Korean total Reasoning tokens: 157.76B				

Table 15: Table 15: Korean and English Datasets Used in Reasoning Mid-Training

Data Filtering Algorithm 1 outlines the procedure for sampling high-difficulty Korean translation seeds from the English Nemotron-Post-V1 dataset. The process consists of two stages, each designed to assess the reasoning difficulty of the samples.

1. **Non-Reasoning Sample Filtering:** We first selected only the samples that were answered incorrectly by both Qwen-30B and Qwen-4B. This approach is based on the intuition that samples demanding complex reasoning are more challenging for instruction-tuned models to solve correctly. Consequently, the samples retained in this stage require detailed reasoning traces and can directly enhance reasoning capability when used for training.
2. **Difficulty Consensus Filtering:** From the Stage 1 candidates, only samples that both Qwen-30B and Qwen-4B labeled as *high-difficulty* were included in the final dataset. While prior studies have often used the length of the reasoning path as a proxy for task difficulty,

this metric becomes unreliable for DeepSeek-family models, which frequently incorporate self-verification steps within their reasoning traces Jung and Jung [2025], Peng et al. [2025]. Moreover, length-based bias can lead to undesirable learning behaviors, such as redundant reasoning patterns or unclear logical structures. To mitigate these issues, we instead utilize model-generated difficulty tags and retain only those samples where both models exhibit a consistent consensus on high difficulty, thereby ensuring the inclusion of truly challenging reasoning data.

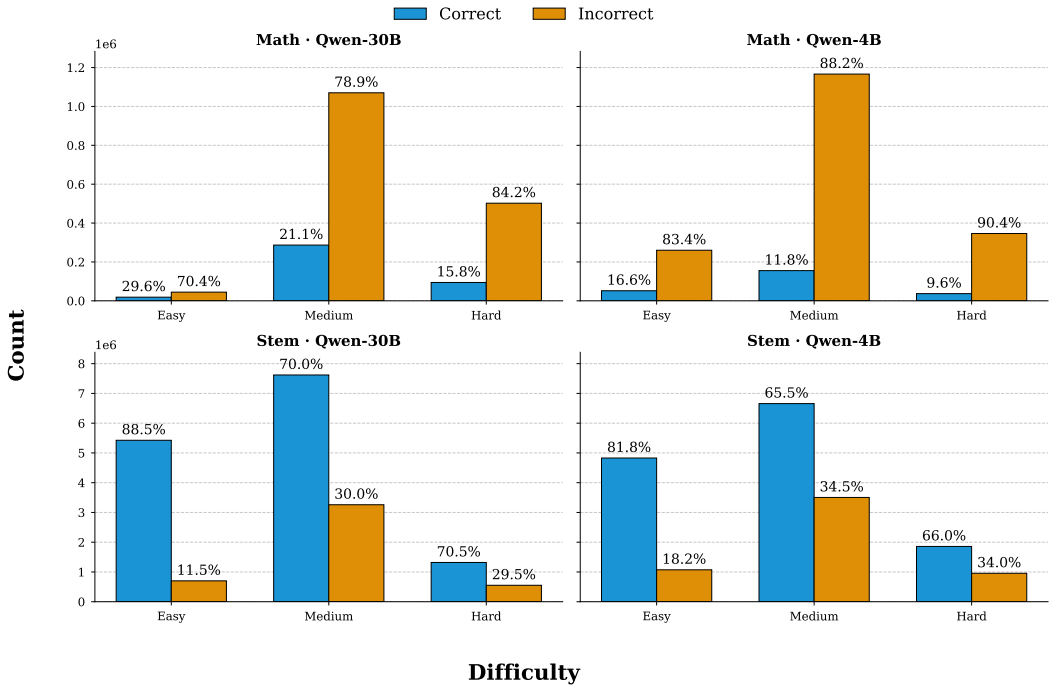


Figure 10: Comparison of accuracy distributions across difficulty levels (Easy, Medium, Hard) as predicted by the two models in the Math and STEM domains. Each bar represents the proportion of correct (blue) and incorrect (orange) samples within each difficulty group. For the STEM domain, correctness was determined using exact match evaluation, given the relatively simple structure of the expected answers. In contrast, for the Math domain, correctness was assessed using Qwen-4B to verify consistency between the model's output and the expected answer. Due to its narrower answer space, the STEM domain exhibits higher overall accuracy. The Code and Chat domains were excluded from the stage 1 filtering process, as objective correctness evaluation is inherently difficult in these domains.

In the code domain, only 48K samples (2.5%) out of 1.90M were retained, resulting in a 97.5% reduction. The math domain decreased from 2.04M to 217K samples (11.0%), and the STEM domain from 20.66M to 196K samples (1.0%), reflecting a 99.0% reduction. Detailed difficulty distributions for math and STEM are shown in Figure 10. For the chat and tool-calling domains, difficulty-based sampling was not applied due to the absence of ground-truth labels.

Based on the selected samples, we constructed the Korean dataset through the following translation procedure. To ensure diversity, we used GPT-OSS-120B and Qwen3-Next-80B-A3B-Instruct in equal proportion (50% each). For the chat and code domains, since the original user queries in Nemotron were sourced from external datasets (lmsys-chat-lm, apps, code contests, open-r1/codeforces, taco), we supplemented incomplete queries using metadata prior to translation.

During translation, we preserved LaTeX expressions, variable names, and function names without modification, and maintained the tone (formal/casual) and style of the original text as closely as possible. For the code domain, we adopted terminology commonly used in Korean programming communities. In the case of tool-calling samples, translations were conducted strictly without simulating agent behavior.

The resulting Korean dataset amounts to approximately 2.8B tokens, measured using tiktoken with the `o200k_base` vocabulary.

The final dataset features multi-step reasoning processes and complex logical structures, providing a foundation for the KORMo model to internalize reasoning ability more effectively and to generalize toward advanced reasoning capabilities. For English data, we used the full dataset without additional filtering, covering the full spectrum of difficulty from simple to challenging samples.

Training Details Following the proposed filtering and translation procedures, we obtained a final reasoning dataset consisting of 150B English tokens and 10B Korean tokens. Each sample was formatted to enclose the reasoning trace within the `<think>` and `</think>` tokens.

During this stage, however, the `<think>` token was excluded from loss computation. This design choice prevents semantic interference in the subsequent SFT stage, where the model will be explicitly trained to treat the `<think>` token as a reasoning trigger signal.

Model	agieval_en	arc_challenge	arc_easy	boolq	copa	winogrande	hellaswag	gpqa_main	Avg.
Stage1	21.78	54.44	82.53	77.34	92.00	72.85	59.05	25.22	60.15
Stage2	28.12	**59.90**	85.65	83.88	**95.00**	**74.98**	59.79	29.46	64.60
Midtrain(Long)	27.60	59.22	**86.11**	**84.04**	**95.00**	74.82	60.11	27.68	**64.57**
Midtrain(Reason)	**28.84**	58.96	85.48	83.46	93.00	74.03	**60.25**	**30.13**	64.27

Table 16: Benchmark performance on major English tasks across training stages.

Model	mmlu	mmlu_global_en	mmlu_pro	mmlu_redux	openbookqa	piqa	social_iqa	commonsense_qa	Avg.
Stage1	56.89	52.54	20.92	58.09	38.60	80.30	52.15	66.42	53.74
Stage2	65.35	61.71	31.32	66.68	38.00	**81.23**	51.89	72.15	58.79
Midtrain(Long)	63.67	61.80	31.47	65.68	**39.20**	80.85	52.15	**72.81**	58.70
Midtrain(Reason)	**67.96**	**63.44**	**40.18**	**69.00**	38.00	81.12	**52.81**	72.24	**60.34**

Table 17: Benchmark performance on major English QA and reasoning tasks across training stages.

5.2.1 Experiment over Mid-training

Comparison of English Benchmark Performance Tables 16 and 17 summarize how the KORMo model's performance on English benchmarks evolved across training stages. The most significant improvement was observed at the Stage 2 phase. On average, the aggregate metrics in Table 16 improved by +4.45 pt (60.15→64.60) compared to Stage 1, and the reasoning-focused benchmarks in Table 17 saw a +5.05-point gain (53.74→58.79). These gains span a broad range of tasks, including ARC-Challenge, BoolQ, and the MMLU series, suggesting that Stage 2 contributes to enhancing the model's generalization capabilities.

Impact of Long-Context Training The Midtrain(Long) stage benefited tasks such as multiple-choice QA and commonsense-based classification by exposing the model to longer contexts and narrative-style inputs. It achieved the highest scores across several benchmarks, including ARC-Easy (86.11), BoolQ (84.04), COPA (95.0), and CommonsenseQA (72.81). However, slight performance drops were observed compared to Stage 2 on more reasoning-intensive tasks such as ARC-Challenge, Winogrande, and GPQA-main. This suggests that while long-context signals may be advantageous for commonsense or surface-level patterns, they have limitations in supporting fine-grained inference or bias control.

Reasoning-Oriented Transfer Learning The Midtrain(Reason) stage led to clear improvements on more challenging, reasoning-centric tasks. Benchmarks such as MMLU(67.96), Global-EN(63.44), MMLU-Pro(40.18), Redux(69.00), and GPQA-main(30.13) all achieved the highest scores in their respective columns, with Social-IQA(52.81) also showing a modest improvement. On average, the Reasoning benchmark suite saw a +1.55pt gain over Stage 2 (58.79→60.34), demonstrating that reasoning-intensive data signals contribute effectively to solving high-difficulty problems. While there was a slight drop in performance on certain tasks like ARC-Challenge and Winogrande, overall performance remained comparable to that of Stage 2.

Model	click	csatqa	haerae	k2_eval	kobest	kobalt	kmmlu	kmmlu_p	kmmlu_r	clinical_qa	mmlu_global	Avg.
Stage1	47.82	28.67	59.30	76.85	69.94	12.71	34.51	26.65	26.83	52.82	41.60	43.43
Stage2	51.43	29.33	62.79	80.79	72.42	17.29	44.38	32.28	36.22	72.92	53.61	50.31
Midtrain(Long)	55.59	30.00	66.73	83.80	71.56	19.00	42.92	34.69	34.94	68.80	53.89	51.08
Midtrain(Reason)	55.29	38.00	68.29	84.49	75.05	22.86	**46.48**	34.51	**37.88**	**77.32**	**55.16**	54.12

Table 18: Model performance on Korean benchmarks.

Analysis Stage2 played a key role in significantly improving the overall base performance, while the subsequent Long and Reason stages provided specialized benefits: the Long stage, which exposes the model to extended contexts and narrative inputs, enhanced commonsense and contextual retrieval ability; the Reason stage, which focuses on high-difficulty reasoning tasks, strengthened precise reasoning ability. Therefore, KORMo's mid-training pipeline suggests that the most effective approach is a two-phase strategy: (1) securing broad foundational capabilities through Stage 2, (2) enhancing commonsense and contextual retrieval ability through Long, and (3) reinforcing precise reasoning ability through Reason. Additionally, since the optimal training stage may vary depending on the target task characteristics, a stage selection strategy aligned with the task domain or a multi-head tuning approach would be more effective in practical applications.

Comparison of Korean Benchmark Performance As summarized in Table 18, KORMo exhibited steady improvements across Korean benchmarks as training progressed. Stage2 yielded the largest performance gain, with a +6.88pt increase in average score compared to Stage 1 (43.43→50.31), particularly improving knowledge-based and reading comprehension tasks such as Clinical-QA(72.92), MMLU-Global(53.61), and K2-Eval(80.79). Subsequently, Midtrain(Long) yielded a modest improvement in the average performance, with notable gains in context-dependent tasks such as Click(55.59) and Haerae(66.73). Lastly, Midtrain(Reasoning) achieved the highest overall performance with an average of 54.12, showing strong improvements on high-difficulty reasoning and knowledge-intensive tasks such as KMMLU(46.48), KMMLU-Redux(37.88), Clinical-QA(77.32), and MMLU-Global(55.16). In summary, Stage 2 establishes strong foundational capabilities, Long enhances performance on context and reading comprehension tasks, and Reason maximizes high-level reasoning capabilities—demonstrating a clear functional specialization across stages in the Korean training pipeline.

Cross-lingual Stage-wise Comparison Interestingly, the performance trends across training stages appear similar for both Korean and English. Stage2 led to the most substantial improvement in overall foundational abilities; the Long stage provided notable gains on context-dependent tasks (e.g., ARC-Easy, BoolQ, Click, Haerae); and the Reason stage achieved the best results on challenging reasoning benchmarks (e.g., MMLU-Pro, GPQA, KMMLU, Clinical-QA).

However, language-specific differences were also observed. In English, the Long stage showed clear improvements on commonsense multiple-choice QA tasks (CommonsenseQA, OpenBookQA), whereas in Korean, the Reason stage led to more dramatic improvements on domain-specific QA tasks such as Clinical-QA.

This suggests that, despite relatively limited training resources, the Korean model was able to reproduce a universal learning progression (Core → Context → Reasoning) through staged training, while also achieving language- and domain-specific improvements.

Therefore, this study demonstrates that stage-wise design can serve not only to boost raw performance in non-English models but also as a fine-tuning tool to compensate for language-specific strengths and weaknesses.

6 Post-training

In this section, we present the *post-training* procedure, the final stage of KORMo's training pipeline. This stage consists of two main components: *Supervised Fine-tuning (SFT)* and a preference learning-based fine-tuning process.

6.1 Supervised Fine-tuning

The SFT stage aims to fine-tune the language model so that it can faithfully understand and execute user instructions. Instead of constructing a new SFT dataset from scratch, we assembled the training data by upsampling data generated in previous stages and incorporating publicly available open-source datasets. However, as emphasized in the LIMA study [Zhou et al., 2023], the performance of SFT is highly sensitive to data quality, which necessitated a rigorous filtering process to ensure the use of high-quality data. To further structure the training, we divided the SFT stage into two phases: **Base SFT**, which focuses on enhancing general reasoning and language capabilities, and **Instruction-Following SFT**, which emphasizes consistent formatting and faithful adherence to user instructions.

Language	Dataset Name	# tokens (M)	Reasoning	Source (Seed)	Synthesizer
		***Base SFT* (6.49B tokens)**			
English	smoltalk_conversations	0.43M	✗	HF-ST2 [Bakouch et al., 2025b]	–
English	smoltalk_system_chats	19.4M	✗	HF-ST2 [Bakouch et al., 2025b]	–
English	smolagents_toolcalling	64.6M	✓	HF-ST2 [Bakouch et al., 2025b]	–
English	nemotron_chat	528.7M	✓	NPT-v1 [Nathawani et al., 2025]	–
English	nemotron_code	714.3M	✓	NPT-v1 [Nathawani et al., 2025]	–
English	nemotron_math	1029.1M	✓	NPT-v1 [Nathawani et al., 2025]	–
English	nemotron_stem	576.3M	✓	NPT-v1 [Nathawani et al., 2025]	–
Korean	Ko-Reasoning	3377.7M	✓	NPT-v1 [Nathawani et al., 2025]	Qwen3-235B-A22B
Multilingual	smoltalk_multilingual	175.1M	✓	HF-ST2 [Bakouch et al., 2025b]	–
		***Instruction Following SFT* (1.53B tokens)**			
English	IF-math	95.4M	✗	MathInst [Yue et al., 2023]	Qwne3-Next-80B-A3B-Instruct
English	IF-math-R	24.2M	✓	MathInst [Yue et al., 2023]	Qwne3-Next-80B-A3B-Thinking
English	IF-stem	18.4M	✗	MoT_science [Face, 2025]	Qwne3-Next-80B-A3B-Instruct
English	IF-stem-R	7.3M	✓	MoT_science [Face, 2025]	Qwne3-Next-80B-A3B-Thinking
English	IF-med	39.3M	✗	PubMed-QA [Jin et al., 2019]	Qwne3-Next-80B-A3B-Instruct
English	IF-med-R	12.9M	✓	PubMed-QA [Jin et al., 2019]	Qwne3-Next-80B-A3B-Thinking
English	WDRM [Boizard et al., 2025]	21.9M	✗	NPT-v1 [Nathawani et al., 2025]	Qwen3-235B-A22B
English	WDRM-R [Boizard et al., 2025]	347M	✓	NPT-v1 [Nathawani et al., 2025]	Qwen3-235B-A22B
English	MAGPIE	474.2M	✗	HF-ST2 [Bakouch et al., 2025b]	Qwne3-Next-80B-A3B-Instruct
English	Magpie-R	39.2M	✓	HF-ST2 [Bakouch et al., 2025b]	Qwne3-Next-80B-A3B-Thinking
Korean	IF-math	31.7M	✗	MathInst [Yue et al., 2023]	Qwne3-Next-80B-A3B-Instruct
Korean	IF-stem	6.7M	✗	MoT_science [Face, 2025]	Qwne3-Next-80B-A3B-Instruct
Korean	IF-med	22.6M	✗	PubMed-QA [Jin et al., 2019]	Qwne3-Next-80B-A3B-Instruct
Korean	Magpie	321.5M	✗	HF-ST2 [Bakouch et al., 2025b]	Qwne3-Next-80B-A3B-Instruct
Korean	Magpie-R	36.2M	✓	HF-ST2 [Bakouch et al., 2025b]	Qwne3-Next-80B-A3B-Thinking
Korean	Ko-Reasoning	31.8M	✓	NPT-v1 [Nathawani et al., 2025]	Qwen3-235B-A22B

Table 19: Overview of the datasets utilized in the supervised fine-tuning (SFT) stage. The table reports token counts, reasoning inclusion flags, and data sources for both English and Korean datasets. **Abbreviations.** HF-ST2 refers to the `HuggingFaceTB/smoltalk2` dataset, NPT-v1 denotes the `Nemotron Post-Training Dataset v1`, NHQ indicates the `Nemotron-HQ`, MoT-science corresponds to the `Mixture-of-Thought` dataset's science subset, and WDRM represents the `When Does Reasoning Matter` project. Underlined datasets indicate those **curated as part of this work**. Together, these datasets comprise a total of 8.02B tokens.

Dataset Filtering. To secure a high-quality SFT dataset, we applied the following procedures:

1. **Deduplication:** We collected data from the Ko-Reasoning dataset and the English Nemotron-Post-Training-Dataset-v1 (chat, code, math, and STEM subsets). Duplicate samples were removed based on the unique identifier (uuid) assigned to each query.

2. **Difficulty-based Sampling:** For Korean data, we employed the Qwen3-30B-A3B-Instruct-2507 model as an evaluator to filter out overly simple samples from the Ko-Reasoning dataset. The model solved STEM, math, and code problems, and its predictions were compared against the correct answers. Correctly answered samples were labeled as *reasoning not required*, while incorrect ones were labeled as *reasoning required*. The final dataset was balanced at a 1:1 ratio between the two categories. For English data, we sampled from the easy, medium, and hard difficulty levels defined in Figure 10, maintaining a balanced distribution across levels.

3. **Length Filtering:** To comply with the model's maximum input length, samples with total sequences exceeding 16,384 tokens were excluded.

Additionally, to ensure broader conversational and functional coverage beyond single-turn question answering, we incorporated data related to *multi-turn* dialogues, *tool calling*, and linguistic diversity from `HuggingFaceTB/smoltalk2`.

BASE SFT Training Strategy. Based on the final dataset, we performed SFT for one epoch to train two model variants:

1. **Reasoning-Enhanced Model:** This variant is designed to explicitly perform reasoning for all queries. During loss computation, all subsequent tokens—including the reasoning span between the `<think>` and `</think>` tokens—were included in the training targets. This setup allows the model to internalize not only the generation of the final answer but also the logical reasoning process leading to it.
2. **Hybrid Model:** This variant allows users to toggle between reasoning and non-reasoning modes. In reasoning mode, the `<think>` block contains an explicit reasoning trace, while in non-reasoning mode, an empty reasoning block (''`<think>\n\n</think>`'') was intentionally inserted during training. Following the previously described difficulty-based sampling strategy, easy samples were assigned to non-reasoning mode and difficult ones to reasoning mode, maintaining a balanced 1:1 ratio overall.

Instruction-Following SFT Training Strategy. In this stage, the model was further fine-tuned to enhance its ability to follow user instructions faithfully and to produce well-structured, contextually consistent responses. The training primarily focused on three aspects: (1) multi-turn dialogue coherence, (2) instruction compliance, and (3) formatting consistency. To achieve this, we constructed instruction-style prompts that required the model to maintain conversational context across turns, interpret and execute user commands accurately, and generate outputs that conformed to predefined response formats (e.g., lists, JSON, Markdown, or structured text). Compared to the Base SFT phase, which emphasized reasoning and general linguistic capability, this stage aimed to refine the model's adherence to task-specific conventions and output structures, thereby improving its usability in real-world interactive scenarios.

During data construction for this phase, particular emphasis was placed on incorporating multi-turn interactions and explicit reasoning paths. For Korean data, English queries were first translated into Korean using the *Qwen3-Next-80B-A3B* model. The translated Korean queries were then re-instructed to the same model to generate corresponding responses, ensuring natural linguistic alignment between queries and answers. To build coherent multi-turn conversations, we adopted the *Magpie* [Xu et al., 2024] methodology, in which the model-generated responses were combined with the `<user>` turn tokens to form extended query-response sequences. In addition, to elicit reasoning traces in Korean, a language-specific signal token was inserted at the beginning of the reasoning span, which guided the model to generate reasoning paths entirely in Korean.

6.2 Optimization on Preference Learning

We constructed two preference learning datasets based on the APO and ORPO frameworks to enhance KORMo's mathematical and reasoning capabilities. For APO, we used prompts from the Nemotron-Post-Train-V2 dataset to generate responses with Qwen3 models of various sizes (0.6B, 4B, 8B, 30B-A3B, and 80B-A3B). Following the SmolLM3 strategy, we treated smaller model outputs as rejected samples and larger model outputs as chosen. Using 100K prompts spanning four domains (chat, STEM, code, and math) we produced 100K responses per model size. Due to GPU constraints, we constructed the dataset without proceeding to model training; training and evaluation results will be released in future work.

7 Experiments

In the preceding sections, we evaluated the experiments and performance at each stage. This chapter reports on the strengths and weaknesses of our proposed method based on a more comprehensive evaluation and compares it with other external models.

7.1 Experiment settings

Applied Models As a hybrid model, KORMo can function in two distinct modes: reasoning and non-reasoning. We performed the evaluation in the non-reasoning mode to maintain consistency with the benchmark models, which operate in a non-reasoning capacity. The reasoning mode's abilities must be strengthened through reinforcement learning, and accordingly, an assessment of the enhanced model will be presented in subsequent work.

Benchmarks To comprehensively evaluate the performance of the KORMo model, we utilized a total of over 26 benchmarks for both English and Korean, covering the domains of *reasoning*, *knowledge*, and *domain-specific* tasks (Table 20).

First, in the **English General Reasoning** category, we evaluate logical reasoning and commonsense inference capabilities using AGIEval, ARC-Challenge/Easy, BoolQ, CommonSenseQA, COPA, HellaSwag, PIQA, Social-IQA, WinoGrande, and OpenBookQA.

The **English Knowledge & Exam-based** category focuses on assessing domain-specific and academic problem-solving skills using MMLU and its extended variants—MMLU-Global, MMLU-Pro, and MMLU-Redux—along with GPQA-Main, which measures performance on advanced scientific questions.

For **Korean General Reasoning**, we employ recently released Korean benchmarks such as CLICK, CSATQA, HAERAE, K2Eval, KoBEST, and KoBALT to measure cultural and linguistic adaptability of the models.

Finally, the **Korean Knowledge & Domain-specific** evaluation includes KMMLU, KMMLU-Pro, KMMLU-Redux, KR-Clinical-QA, and MMLU-Global (KR). These benchmarks assess the models' ability to comprehend and respond to Korean-domain expert questions in fields such as medicine and law. Among them, KR-Clinical-QA is the most recently released benchmark and is included to mitigate potential data contamination that might occur with earlier public datasets, ensuring a more objective measurement of model performance.

All benchmarks are evaluated under a 5-shot prompting setup (4-shot for GPQA-Main), and accuracy is used as the unified evaluation metric. This configuration aims to verify whether KORMo achieves balanced reasoning and knowledge competence across multilingual and multi-domain environments.

Lastly, for the **instruction-tuned models**, we evaluate models that have undergone the same instruction tuning procedure. To assess instruction-following ability, we use three representative *LLM-as-a-Judge* benchmarks, where GPT-4o serves as a consistent evaluation model for all systems.

Evaluation Prompt We adopted the standard evaluation prompts proposed by OLMo 2 (as shown in Table 20). To assess instruction-following capabilities, we evaluated all models under the same conditions using GPT-4o and the same set of prompts.

prompt MMLU

```
Q: {question}
Put your answer within \boxed{}.
```

Figure 11: Prompt for evaluating MMLU

7.2 Base Model Evaluation

The evaluation of base models aims to assess the foundational capabilities of language understanding, reasoning, and mathematical skills prior to instruction tuning. Accordingly, we compare only the base versions of models, without any instruction tuning applied, to ensure a fair evaluation. Among recently released multilingual models, only Qwen3, Gemma3, and LLaMA3.1 offer publicly available base versions. For Korean, Kanana1.5 is the sole model with a base release. Thus, this evaluation is designed to isolate the effects of pretraining, excluding any influence from instruction tuning or RLHF. Meanwhile, Fully-Open Model(FOMs) such as SmolLM3 and OLMo2 also release base versions; however, these are primarily trained on English and other Indo-European languages.

Name	Language	Prompting	Metric
General Reasoning (English)			
AGIEval Zhong et al. [2023]	English	5-shot	accuracy
ARC-Challenge Clark et al. [2018a]	English	5-shot	accuracy
ARC-Easy Clark et al. [2018a]	English	5-shot	accuracy
BoolQ Clark et al. [2019b]	English	5-shot	accuracy
CommonSenseQA Talmor et al. [2019]	English	5-shot	accuracy
COPA Roemmele et al. [2011]	English	5-shot	accuracy
HellaSwag Zellers et al. [2019b]	English	5-shot	accuracy
PIQA Bisk et al. [2020a]	English	5-shot	accuracy
Social-IQA Sap et al. [2019]	English	5-shot	accuracy
WinoGrande Sakaguchi et al. [2021a]	English	5-shot	accuracy
OpenBookQA Mihaylov et al. [2018]	English	5-shot	accuracy
Knowledge & Exam-based (English)			
MMLU Hendrycks et al. [2020a]	English	5-shot	accuracy
MMLU-Global (EN) Hendrycks et al. [2020a]	English	5-shot	accuracy
MMLU-Pro Hendrycks et al. [2020a]	English	5-shot	accuracy
MMLU-Redux Hendrycks et al. [2020a]	English	5-shot	accuracy
GPQA-Main Rein et al. [2024]	English	4-shot	accuracy
General Reasoning (Korean)			
CLICK Kim et al. [2024]	Korean	5-shot	accuracy
CSATQA[22]	Korean	5-shot	accuracy
HAERAE Son et al. [2023b]	Korean	5-shot	accuracy
K2Eval[23]	Korean	5-shot	accuracy
KoBEST Jang et al. [2022a]	Korean	5-shot	accuracy
KoBALT Shin et al. [2025]	Korean	5-shot	accuracy
Knowledge & Domain-specific (Korean)			
KMMLU Son et al. [2025b]	Korean	5-shot	accuracy
KMMLU-Pro Hong et al. [2025]	Korean	5-shot	accuracy
KMMLU-Redux Hong et al. [2025]	Korean	5-shot	accuracy
KR-Clinical-QA [24]	Korean	5-shot	accuracy
MMLU-Global (KR) Singh et al. [2025]	Korean	5-shot	accuracy
Instruction-Following			
MT-Bench	English	LLM-as-Judge(GPT-4o)	LLM score
Ko-MT-Bench	Korean	LLM-as-Judge(GPT-4o)	LLM score
LogicKor	Korean	LLM-as-Judge(GPT-4o)	LLM score

Table 20: **Evaluation benchmarks used in Table 22.** Each benchmark is categorized by domain and language, with its prompting setup (shot) and evaluation metric.

Overall Trends When evaluated in its *base* form (before instruction tuning), KORMo-10B scored 64.2 on English and 58.2 on Korean benchmarks. This indicates stable performance compared to other models in the Fully-Open category. In English, it performed on par with large open models such as OLMo2-13B (64.2 vs. 65.3), while in Korean, it scored approximately 4–8% lower than multilingual LLMs including KANANA-8B, Qwen3-8B, and Gemma3-12B. Given its relatively modest pretraining corpus of 2.9T tokens, these results highlight KORMo-10B's high language modeling efficiency.

English Benchmarks: Robust Reasoning Ability KORMo consistently demonstrated strong performance in general reasoning (ARC, BoolQ, HellaSwag) and commonsense inference tasks (PIQA, Social-IQA, WinoGrande). While larger 12B and 13B models generally achieved higher scores, the margin was relatively narrow (2–7%). KORMo also consistently outperformed other mid-sized fully open models. This suggests that KORMo was able to acquire the essential patterns of English reasoning, even without exposure to the wide domain coverage typically seen in large multilingual models. On the other hand, it showed slightly lower performance on more knowledge-intensive tasks such as MMLU-Pro and GPQA, suggesting that KORMo's strength lies more in general reasoning and linguistic consistency than in fine-grained factual precision.

Korean General Reasoning: High Adaptability in Native Language KORMo exhibited balanced performance across both English and Korean in comprehensive reasoning benchmarks such as K2-

	Fully-Open Model				Open-Weight Model (Multilingual)				
Benchmark **Tokens**	kormo-10b (2.9T)	smolLM3-3b (10.7T)	olmo2-7b (4T)	olmo2-13b (5.5T)	kanana1.5-8b (3.2T)	qwen3-8b (36T)	llama3.1-8b (15T)	gemma3-4b (4T)	gemma3-12b (12T)
English Benchmarks									
arc_challenge	58.96	55.55	59.13	61.01	56.48	63.82	54.61	53.58	63.82
arc_easy	85.48	83.21	85.06	86.57	82.74	87.50	84.01	82.83	87.37
boolq	83.46	82.17	84.50	86.48	84.53	87.71	81.87	80.70	86.61
copa	93.00	91.00	92.00	93.00	88.00	92.00	93.00	89.00	95.00
gpqa_main	30.13	26.79	26.34	29.24	29.24	30.13	23.44	30.13	35.71
hellaswag	60.25	56.78	61.52	65.02	59.93	59.54	60.96	57.56	63.67
mmlu	67.96	61.37	62.81	66.85	63.73	76.95	65.03	59.60	73.58
mmlu_global	63.44	57.52	59.88	63.99	60.21	75.05	61.30	57.23	70.23
mmlu_pro	40.18	34.94	27.29	32.50	34.93	56.58	36.23	27.79	37.07
mmlu_redux	69.00	62.95	63.53	68.37	65.88	78.19	65.86	60.86	75.25
openbookqa	39.00	36.40	39.00	39.60	36.80	39.20	39.00	37.00	40.20
piqa	81.12	78.45	80.79	82.64	80.30	79.05	80.90	79.49	82.59
social_iqa	52.81	50.72	55.89	57.57	57.01	56.96	53.12	51.84	56.45
winogrande	74.03	73.32	77.03	81.69	73.32	77.03	77.74	72.93	80.51
English Avg.	**64.20**	**60.80**	**62.48**	**65.32**	**62.36**	**68.55**	**62.65**	**60.04**	**67.72**
Korean Benchmarks									
click	55.29	46.97	37.79	41.80	62.76	60.70	49.22	49.62	62.21
csatqa	38.00	26.67	19.33	24.67	44.67	52.00	28.67	28.67	31.33
haerae	68.29	55.82	31.62	37.58	80.75	67.19	53.25	60.68	74.34
k2_eval	84.89	75.23	49.54	63.43	84.72	84.72	76.62	76.39	85.42
kobest	75.05	69.13	57.27	59.02	81.93	80.05	70.55	69.33	77.70
kobalt	22.86	15.86	11.43	13.14	26.29	26.57	17.43	15.57	23.86
kmmlu	46.48	38.52	33.05	31.24	48.86	56.93	40.75	39.84	51.60
mmlu_global	55.16	44.15	34.00	36.95	52.65	61.95	46.34	46.33	59.68
kr_clinical_qa	77.32	53.97	48.33	46.22	65.84	80.00	63.54	60.00	77.22
Korean Avg.	**58.15**	**47.37**	**35.82**	**39.34**	**60.94**	**63.35**	**49.60**	**49.60**	**60.37**

Table 21: Performance comparison of Fully-Open and Open-Weight multilingual models on English and Korean benchmarks. All scores represent accuracy, and averages are simple arithmetic means. Numbers in parentheses below model names indicate training tokens (in trillions).

Eval, a general-purpose Korean reasoning task. This suggests that the quality of Korean data and its proportion in pretraining were effective. However, performance was relatively lower on tasks like KOBALT, which require fine-grained lexical and semantic discrimination. This suggests that additional post-training may be necessary to better capture subtle linguistic distinctions.

We also observed that Korean performance generally improved with a higher proportion of Korean data. Compared to the KANANA-1.5 model, KORMo scored 1.84 points higher in English, but 2.79 points lower in Korean. Considering that KANANA reportedly uses approximately 10% Korean data, this result reflects the effect of KORMo's lower Korean data ratio, estimated at around 5.6%.

Korean Knowledge & Domain Tasks: Specialized Strengths On the Clinical-QA benchmark, which focuses on domain-specific knowledge, KORMo showed notable strength in **clinical and practical** tasks. This suggests that the model effectively internalized medical and commonsense data presented in real-world QA formats during training. On the other hand, in the KMMLU suite, which focuses on academic and advanced knowledge-based tasks, KORMo showed weaker performance than large multilingual models. This indicates a pretraining bias toward general reasoning rather than fine-grained domain-specific knowledge. The result also implies a lack of exposure to high-quality expert data or explanatory QA formats, likely due to limitations in generating advanced professional content within the augmented Korean datasets.

Cross-lingual Patterns and Efficiency The average score gap between English and Korean for KORMo was around 6 points, demonstrating a level of balance comparable to large-scale multilingual models such as Qwen and Gemma. While KANANA, which is optimized for Korean, showed a much smaller gap of about 1 point, its English performance was relatively limited. In contrast, KORMo demonstrated **stable performance across both languages**, achieving a well-balanced trade-off between linguistic coverage and general reasoning ability. This outcome underscores the effectiveness of our data design, enabled by fine-grained control over bilingual proportions within a limited token budget.

Insights and Future Directions In summary, KORMo demonstrates (1) robustness on English general reasoning, (2) strong adaptability on Korean general reasoning and practical QA, and (3)

	Fully-Open Model				Open-Weight Model (Multilingual)				
Benchmark **Tokens**	kormo-10b (2.9T)	smolLM3-3b (10.7T)	olmo2-7b (4T)	olmo2-13b (5.5T)	kanana1.5-8b (3.2T)	qwen3-8b (36T)	llama3.1-8b (15T)	exaone3.5-8b* (12T)	gemma3-12b (12T)
MT-Bench	8.32	7.15	7.32	7.64	8.45	8.70	6.32	8.15	8.70
KO-MT-Bench	8.54	-	-	-	8.02	8.16	4.27	8.13	8.51
Logickor	8.96	-	-	-	8.94	8.63	6.45	9.20	8.46
Average	**8.61**	-	-	-	**8.47**	**8.50**	**5.68**	**8.49**	**8.56**

Table 22: Benchmark performance (MT-Bench, KO-MT-Bench, Logickor) of Fully-Open and Open-Weight multilingual language models. Scores are on a 10-point scale, with averages computed as simple arithmetic means. Numbers in parentheses indicate training tokens (in trillions). All evaluations were automatically scored using GPT-4o. *Since Exaone4 is available in 1B and 32B sizes, we conducted the comparison using the 8B model.

superior token efficiency. In contrast, we observe relative weaknesses on (a) high-difficulty specialist knowledge benchmarks (e.g., MMLU-Pro, the KMMLU family) and (b) lexical semantic discrimination tasks (e.g., KOBALT), which remain targets for improvement. To address these gaps, we plan to explore (i) high-quality, domain-knowledge–oriented mid-training, (ii) Korean-centric contrastive fine-tuning, and (iii) post-training strategies that incorporate diverse supervision formats (e.g., explanatory QA and chain-of-thought). Overall, our results provide empirical evidence that, for LLMs, data quality and linguistic balance matter more than sheer token volume in determining generality and efficiency.

7.3 Evaluation of SFT Models

We believe KORMo was trained on a larger volume of Korean instruction data than any of the models it is compared against. This was a deliberate design decision, as we focused on the model's capability to respond suitably to various Korean instructions instead of performing well on MMLU-style multiple-choice questions. Our focus was on enhancing the practical, real-world performance as perceived by users.

Experimental Setup and Preliminary The present experiment was conducted with models limited to those that have completed **only the instruction tuning stage**. This means we are comparing their linguistic and instruction execution capabilities prior to any further reasoning enhancement via reinforcement learning (such as RLHF, GRPO, or APO). Such a configuration is intended to assess a model's capacity for generating natural and coherent responses based solely on supervised data. The selected benchmarks (MT-Bench, KO-MT-Bench, and Logickor) encompass tasks focused on general dialogue, Korean-specific dialogue, and logical inference, respectively, and collectively reflect the models' linguistic fluency, fidelity to instructions, and logical coherence.

Instruction Following Ability (MT-Bench). MT-Bench and LogicKor serve as multi-domain benchmarks for evaluating instruction-following capabilities in English and Korean, designed to offer a holistic assessment of response quality and consistency in instruction-tuned models. The results of our comparative analysis of mean scores reveal that KORMo-10B achieved the highest average score (8.61). This places its performance in close equivalence to commercial-grade models like Gemma3-12B (8.56) and Qwen3-8B (8.50). Such a result suggests that KORMo successfully captured the expressiveness and structural variety of the instruction corpus despite being trained with comparatively fewer tokens (2.9T) and resources.

The Effect of Language Distribution in Instruction Tuning Data on Model Efficiency. When comparing the mean scores for English (MT-Bench) against Korean benchmarks (KO-MT-Bench, Logickor), KORMo demonstrates superior capability in Korean, achieving an average score of 8.75, in contrast to 8.32 for English. This finding indicates that KORMo has thoroughly incorporated its Korean-focused instruction fine-tuning dataset. We infer that this is because KORMo was trained on a more substantial amount of Korean instruction data than the other models under consideration.

Conclusion. First, KORMo achieved performance similar to large multilingual models in an instruction-only setup, which we credit to effectively using data quality and instructional diversity. Second, the KO-MT-Bench and Logickor benchmarks show that KORMo demonstrates greater

robustness in Korean conversational contexts and logical inference tasks. Third, achieving an 8.6 MT-Bench score without reinforcement learning suggests that instruction tuning alone is enough for understanding user instructions and generating clear, logical responses. Finally, the main advantage of KORMo is not just its multilingual ability, but its effective knowledge alignment through high-quality bilingual instruction fine-tuning. For the subsequent reinforcement learning stage, our objective is to expand upon this base by extending reasoning chains and reinforcing multi-hop consistency, thus transitioning the model from its instruction-tuned capabilities to those of a reasoning-capable agent.

7.4 Evaluation of Reinforcement Learning Models

As a final step, we will further strengthen the model's reasoning capabilities by incorporating mathematics and logical reasoning data generated using our previously introduced, custom-developed APO and GRPO frameworks.

8 Conclusion

This paper introduced **KORMo**, the first fully open bilingual Korean-English LLM developed primarily with synthetic data. Through extensive quantitative analysis, we demonstrated that synthetic corpora can effectively replace large-scale human-curated data when designed with careful linguistic balance and stylistic diversity. Across both pretraining and instruction-tuning stages, KORMo exhibits stable convergence, strong multilingual generalization, and high reasoning consistency-comparable to leading open-weight models considering the small number of trained tokens.

Empirical evaluation across over various benchmarks revealed several key insights. First, KORMo maintains competitive English reasoning ability while achieving superior Korean instruction-following and logical coherence. Second, task-specific results (e.g., KR-Clinical-QA, Logickor) highlight the complementary strengths of synthetic data: efficiency in generalized reasoning, yet room for improvement in fine-grained expert knowledge. Third, the language mixture and tokenizer analyses confirm that balanced bilingual compression improves token efficiency and stability, providing a replicable guideline for future FOM builders.

Importantly, KORMo validates that **synthetic data is not only viable but also scalable** as a principal resource for non-English FOMs. This finding challenges the prevailing assumption that synthetic augmentation leads to model collapse or cultural drift. Instead, with transparent preprocessing, curriculum control, and open evaluation, synthetic data can serve as a sustainable foundation for reproducible multilingual research.

In conclusion, KORMo bridges the gap between open-weight multilingual models and fully open, reproducible pipelines. By publicly releasing the entire data and training process, it enables the community to extend and audit every component of the LLM lifecycle. Future work will expand upon this foundation by (i) incorporating reasoning-oriented reinforcement learning, (ii) exploring multilingual generalization beyond Korean–English, and (iii) establishing standardized evaluation suites for synthetic-data–driven language modeling. Through this effort, KORMo aims to advance the broader movement toward *transparent, reproducible, and inclusive foundation model research*.

Acknowledgments

Special Thanks. We sincerely thank *한승준, 석주영, 최승택, 김규석,* and *신재민* from Trillion Labs for generously sharing their early experience in model design. We are also grateful to *한승윤, 이준명, 고창건, 황의준* and *황태호* from the KAIST NLPCL Lab and *송서현* from Seoultech for their insightful discussions and valuable feedback during the model study phase. We would like to express our gratitude to *김기현* from LG U+ for contributing high-quality Korean data to this project. Finally, we deeply appreciate the support and efforts of the AWS and KETI teams for their assistance in ensuring the smooth execution of this work.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2025-02653113, High-Performance Research AI Computing Infrastructure Support at the 2 PFLOPS Scale)

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. 2023. URL https://arxiv.org/abs/2305.13245.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go mad. 2023.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, et al. Tokenizer choice for llm training: Negligible or crucial? pages 3907–3924, 2024.

Ansar Aynetdinov and Alan Akbik. Pre-training curriculum for multi-token prediction in language models, 2025. URL https://arxiv.org/abs/2505.22757.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. 2022. URL https://arxiv.org/abs/2204.05862.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patino, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, et al. Smollm3: smol, multilingual, long-context reasoner, 2025a.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3, 2025b.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. 2020. URL https://arxiv.org/abs/2004.05150.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Cosmopedia, 2024. URL https://huggingface.co/datasets/HuggingFaceTB/cosmopedia.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020a.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439, 2020b.

Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Kevin El-Haddad, Céline Hudelot, and Pierre Colombo. When does reasoning matter? a controlled study of reasoning's contribution to model performance, 2025. URL https://arxiv.org/abs/2509.22193.

Alexander Bukharin, Shiyang Li, Zhengyang Wang, Jingfeng Yang, Bing Yin, Xian Li, Chao Zhang, Tuo Zhao, and Haoming Jiang. Data diversity matters for robust instruction tuning. 2024. URL https://arxiv.org/abs/2311.14736.

Yekun Chai, Yewei Fang, Qiwei Peng, and Xuhong Li. Tokenization falling short: On subword robustness in large language models. 2024. URL https://arxiv.org/abs/2406.11687.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I. Abdin. On the diversity of synthetic data and its impact on training large language models. 2024. URL https://arxiv.org/abs/2410.15226.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. 2019a. URL https://arxiv.org/abs/1905.10044.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. 2019b. URL https://arxiv.org/abs/1905.10044.

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

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. 2018b. URL https://arxiv.org/abs/1803.05457.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. 2023. URL https://arxiv.org/abs/2307.08691.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/67d57c32e20fd0a7a302cb81d36e40d5-Paper-Conference.pdf.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. 2025a. URL https://arxiv.org/abs/2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025b. URL https://arxiv.org/abs/2412.19437.

Zican Dong, Junyi Li, Jinhao Jiang, Mingyu Xu, Wayne Xin Zhao, Bingning Wang, and Weipeng Chen. Longred: Mitigating short-text degradation of long-context large language models via restoration distillation. 2025. URL https://arxiv.org/abs/2502.07365.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. 2024. URL https://arxiv.org/abs/2402.01306.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. 2020. URL https://arxiv.org/abs/2101.00027.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively). pages 7376–7399, July 2025a. doi: 10.18653/v1/2025.acl-long.366. URL https://aclanthology.org/2025.acl-long.366/.

Tianyu Gao, Alexander Wettig, Howard Yen, and Danqi Chen. How to train long-context language models (effectively). In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7376–7399, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.366. URL https://aclanthology.org/2025.acl-long.366/.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. 2020. URL https://arxiv.org/abs/2009.11462.

Matthias Gerstgrasser, Rylan Schaeffer, et al. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. In *COLM*, 2024. URL https://openreview.net/forum?id=5B2K4LRgmz.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction, 2024. URL https://arxiv.org/abs/2404.19737.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily

Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models. 2024. URL `https://arxiv.org/abs/2407.21783`.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. 2023. URL `https://arxiv.org/abs/2306.11644`.

Sungjun Han, Juyoung Suk, Suyeong An, Hyungguk Kim, Kyuseok Kim, Wonsuk Yang, Seungtaek Choi, and Jamin Shin. Trillion 7b technical report. *arXiv preprint arXiv:2504.15431*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. 2020a.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. 2020b.

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

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. 2024. URL `https://arxiv.org/abs/2403.07691`.

Seokhee Hong, Sunkyoung Kim, Guijin Son, Soyeon Kim, Yeonjung Hong, and Jinsik Lee. From kmmlu-redux to kmmlu-pro: A professional korean benchmark suite for llm evaluation, 2025. URL https://arxiv.org/abs/2507.08924.

Myeongjun Jang, Dohyung Kim, Deuk Sin Kwon, and Eric Davis. Kobest: Korean balanced evaluation of significant tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3697–3708, 2022a.

Myeongjun Jang, Dohyung Kim, Deuk Sin Kwon, and Eric Davis. Kobest: Korean balanced evaluation of significant tasks. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3697–3708, 2022b.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*, 2019.

Jeesu Jung and Sangkeun Jung. Reasoning steps as curriculum: Using depth of thought as a difficulty signal for tuning llms. *arXiv preprint arXiv:2508.18279*, 2025.

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

Eunsu Kim, Juyoung Suk, Philhoon Oh, Haneul Yoo, James Thorne, and Alice Oh. Click: A benchmark dataset of cultural and linguistic intelligence in korean, 2024. URL https://arxiv.org/abs/2403.06412.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. 2018. URL https://arxiv.org/abs/1808.06226.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. 2017. URL https://arxiv.org/abs/1704.04683.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 14200–14282. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/19e4ea30dded58259665db375885e412-Paper-Datasets_and_Benchmarks_Track.pdf.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. 2023. URL https://arxiv.org/abs/2309.05463.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. 2022. URL https://arxiv.org/abs/2109.07958.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the ACL*, 2024. URL https://aclanthology.org/2024.tacl-1.9/.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On llms-driven synthetic data generation, curation, and evaluation: A survey. 2024. URL https://arxiv.org/abs/2406.15126.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14867–14875, 2021.

Somesh Mehra, Javier Alonso Garcia, and Lukas Mauch. On multi-token prediction for efficient llm inference, 2025. URL https://arxiv.org/abs/2502.09419.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. Olmoe: Open mixture-of-experts language models. *arXiv preprint arXiv:2409.02060*, 2024.

Dhruv Nathawani, Igor Gitman, Somshubra Majumdar, Evelina Bakhturina, Ameya Sunil Mahabaleshwarkar, , Jian Zhang, and Jane Polak Scowcroft. Nemotron-Post-Training-Dataset-v1, 2025. URL https://huggingface.co/datasets/nvidia/Nemotron-Post-Training-Dataset-v1.

Nvidia, :, Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, Aleksander Ficek, Denys Fridman, Shaona Ghosh, Boris Ginsburg, Igor Gitman, Tomasz Grzegorzek, Robert Hero, Jining Huang, Vibhu Jawa, Joseph Jennings, Aastha Jhunjhunwala, John Kamalu, Sadaf Khan, Oleksii Kuchaiev, Patrick LeGresley, Hui Li, Jiwei Liu, Zihan Liu, Eileen Long, Ameya Sunil Mahabaleshwarkar, Somshubra Majumdar, James Maki, Miguel Martinez, Maer Rodrigues de Melo, Ivan Moshkov, Deepak Narayanan, Sean Narenthiran, Jesus Navarro, Phong Nguyen, Osvald Nitski, Vahid Noroozi, Guruprasad Nutheti, Christopher Parisien, Jupinder Parmar, Mostofa Patwary, Krzysztof Pawelec, Wei Ping, Shrimai Prabhumoye, Rajarshi Roy, Trisha Saar, Vasanth Rao Naik Sabavat, Sanjeev Satheesh, Jane Polak Scowcroft, Jason Sewall, Pavel Shamis, Gerald Shen, Mohammad Shoeybi, Dave Sizer, Misha Smelyanskiy, Felipe Soares, Makesh Narsimhan Sreedhar, Dan Su, Sandeep Subramanian, Shengyang Sun, Shubham Toshniwal, Hao Wang, Zhilin Wang, Jiaxuan You, Jiaqi Zeng, Jimmy Zhang, Jing Zhang, Vivienne Zhang, Yian Zhang, and Chen Zhu. Nemotron-4 340b technical report. 2024. URL https://arxiv.org/abs/2406.11704.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, et al. 2 olmo 2 furious. *arXiv preprint arXiv:2501.00656*, 2024.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 olmo 2 furious. 2025. URL https://arxiv.org/abs/2501.00656.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data only. *Advances in Neural Information Processing Systems*, 36:79155–79172, 2023.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024. URL https://arxiv.org/abs/2406.17557.

Keqin Peng, Liang Ding, Yuanxin Ouyang, Meng Fang, and Dacheng Tao. Revisiting overthinking in long chain-of-thought from the perspective of self-doubt. *arXiv preprint arXiv:2505.23480*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95, 2011.

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Margetts, and Janet Pierrehumbert. Hatecheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.4. URL http://dx.doi.org/10.18653/v1/2021.acl-long.4.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. volume 64, pages 99–106. ACM New York, NY, USA, 2021a.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. volume 64, pages 99–106. ACM New York, NY, USA, 2021b.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. 2019. URL https://arxiv.org/abs/1904.09728.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? *Advances in neural information processing systems*, 36:55565–55581, 2023.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. 2016. URL https://arxiv.org/abs/1508.07909.

Skyler Seto, Maartje Ter Hoeve, Richard He Bai, Natalie Schluter, and David Grangier. Training bilingual LMs with data constraints in the targeted language. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19096–19122, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.977. URL https://aclanthology.org/2025.findings-acl.977/.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37:68658–68685, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. 2024. URL https://arxiv.org/abs/2402.03300.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. 2019. URL https://arxiv.org/abs/1911.02150.

Noam Shazeer. Glu variants improve transformer. 2020. URL https://arxiv.org/abs/2002.05202.

Hyopil Shin, Sangah Lee, Dongjun Jang, Wooseok Song, Jaeyoon Kim, Chaeyoung Oh, Hyemi Jo, Youngchae Ahn, Sihyun Oh, Hyohyeong Chang, Sunkyoung Kim, and Jinsik Lee. Kobalt: Korean benchmark for advanced linguistic tasks, 2025. URL https://arxiv.org/abs/2505.16125.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023a.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023b.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.919. URL https://aclanthology.org/2025.acl-long.919/.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. 2024. URL https://arxiv.org/abs/2402.00159.

Guijin Son, Hanwool Lee, Suwan Kim, Huiseo Kim, Jaecheol Lee, Je Won Yeom, Jihyu Jung, Jung Woo Kim, and Songseong Kim. Hae-rae bench: Evaluation of korean knowledge in language models. 2023a.

Guijin Son, Hanwool Lee, Suwan Kim, Huiseo Kim, Jaecheol Lee, Je Won Yeom, Jihyu Jung, Jung Woo Kim, and Songseong Kim. Hae-rae bench: Evaluation of korean knowledge in language models. 2023b.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. KMMLU: Measuring massive multitask language understanding in Korean. pages 4076–4104, April 2025a. doi: 10.18653/v1/2025.naacl-long.206. URL https://aclanthology.org/2025.naacl-long.206/.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. KMMLU: Measuring massive multitask language understanding in Korean. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4076–4104, Albuquerque, New Mexico, April 2025b. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.206. URL https://aclanthology.org/2025.naacl-long.206/.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. *arXiv preprint arXiv:2412.02595*, 2024a.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc: Transforming common crawl into a refined long-horizon pretraining dataset. 2025. URL https://arxiv.org/abs/2412.02595.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024b.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 2019. URL https://arxiv.org/abs/1811.00937.

Qwen Team. Qwen3 technical report. 2025a. URL https://arxiv.org/abs/2505.09388.

Qwen Team. Qwen3 technical report, 2025b. URL https://arxiv.org/abs/2505.09388.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6761–6774, 2024a.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. 2023. URL https://arxiv.org/abs/2212.10560.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024b.

Yudong Wang, Zixuan Fu, Jie Cai, Peijun Tang, Hongya Lyu, Yewei Fang, Zhi Zheng, Jie Zhou, Guoyang Zeng, Chaojun Xiao, Xu Han, and Zhiyuan Liu. Ultra-fineweb: Efficient data filtering and verification for high-quality llm training data. 2025a. URL https://arxiv.org/abs/2505.05427.

Yudong Wang, Zixuan Fu, Jie Cai, Peijun Tang, Hongya Lyu, Yewei Fang, Zhi Zheng, Jie Zhou, Guoyang Zeng, Chaojun Xiao, et al. Ultra-fineweb: Efficient data filtering and verification for high-quality llm training data. *arXiv preprint arXiv:2505.05427*, 2025b.

Maurice Weber, Dan Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. Redpajama: an open dataset for training large language models. *Advances in neural information processing systems*, 37: 116462–116492, 2024.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. 2022. URL https://arxiv.org/abs/2206.07682.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim

Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. Bloom: A 176b-parameter open-access multilingual language model. 2023. URL `https://arxiv.org/abs/2211.05100`.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture.

In *International conference on machine learning*, pages 10524–10533. PMLR, 2020.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*, 2024.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41/.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

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

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

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

Yu Zhao, Yuanbin Qu, Konrad Staniszewski, Szymon Tworkowski, Wei Liu, Piotr Miłoś, Yuxiang Wu, and Pasquale Minervini. Analysing the impact of sequence composition on language model pre-training. page 7897–7912, 2024. doi: 10.18653/v1/2024.acl-long.427. URL http://dx.doi.org/10.18653/v1/2024.acl-long.427.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. 2023. URL https://arxiv.org/abs/2304.06364.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

Jingwei Zuo, Maksim Velikanov, Ilyas Chahed, Younes Belkada, Dhia Eddine Rhayem, Guillaume Kunsch, Hakim Hacid, Hamza Yous, Brahim Farhat, Ibrahim Khadraoui, et al. Falcon-h1: A family of hybrid-head language models redefining efficiency and performance. *arXiv preprint arXiv:2507.22448*, 2025.