

Optimizing Multilingual BabyLM: The Roles of Tokenization, Data Packing, and Curriculum Learning

Fuming Yang¹ Hsuan Lien¹ Kai-Hao Yang¹
Lukas Edman^{1,2} Alexander Fraser^{1,2,3}

¹School of Computation, Information and Technology, TU Munich

²Munich Center for Machine Learning

³Munich Data Science Institute

go93pub@mytum.de, go46lic@mytum.de, go82qeq@mytum.de, lukas.edman@tum.de

Abstract

Training multilingual language models under strict data constraints remains a challenging problem. This paper presents our submission to the BabyLM 2026 Multilingual Track, focusing on English, Dutch, and Chinese. Three key challenges arise under the BabyLM constraints: efficiently sharing limited model capacity, maximizing sequence utilization, and allocating parameters efficiently under a fixed vocabulary budget. To address these challenges, we propose a three-stage curriculum learning strategy that dynamically adjusts language mixing ratios throughout training. Furthermore, we investigate data packing strategies, comparing naive chunking with document wrapping to maximize the learning signal within a strict 100M-token budget. Finally, we design a custom BPE tokenizer with an asymmetric sampling strategy. By reducing the size of the embedding matrix, the saved parameters are reallocated to the Transformer backbone, enabling highly efficient GPT-2 and BERT models. Experimental results show that combining our curriculum strategy with document packing consistently improves GPT-2 training convergence and achieves highly competitive multilingual performance.

1 Introduction

The BabyLM 2026 Multilingual Track (Choshen et al., 2026b) challenges participants to train multilingual language models under a strict data budget. Unlike conventional large-scale language models trained on billions of tokens, the BabyLM Challenge restricts the training corpus to encourage data-efficient language modeling; model size itself is not formally restricted, but participants typically keep models compact, since prior BabyLM findings suggest that scaling parameters further yields limited benefit under such constrained data budgets. The goal is to investigate how effectively language models can learn under these resource limitations.

1.1 Challenge

When training a multilingual language model under strict data constraints, we consider three main problems.

First, a significant structural bottleneck arises from the tokenizer’s vocabulary size. A multilingual tokenizer must support subwords in the Latin alphabet for English and Dutch while simultaneously covering thousands of unique Chinese characters. This necessitates a massive vocabulary, which in turn requires a highly parameterized embedding matrix. If the overall model is kept compact for training efficiency, a larger vocabulary allocates a greater proportion of the model parameters to the embedding layer. Consequently, fewer parameters remain available for the Transformer layers, potentially limiting the model’s representation capacity.

Second, efficiently utilizing the limited data budget during sequence formatting is highly challenging. When mixing documents of varying lengths, naive truncation or padding wastes the strict 100M-token budget on useless tokens or splits semantic contexts. Efficient data packing strategies are required to maximize the learning signal extracted from the available context window, ensuring the model shares its limited capacity effectively across languages with diverse linguistic characteristics.

Third, determining the optimal data mixing strategy is highly complex due to inherent language differences. English and Dutch share the Latin alphabet but differ in syntactic structures, while Chinese uses a character-based writing system with fundamentally different linguistic properties. As a result, these languages exhibit different learning behaviors during training. Determining how to dynamically adjust the allocation of the limited training budget across languages over time—rather than using a static mix—is therefore a key challenge.

1.2 Our Contribution

To solve the challenges mentioned above, we built a highly efficient multilingual language model for the BabyLM 2026 Multilingual Track. Beyond the engineering contributions listed below, our experiments also yield a concrete scientific finding: unlike much of the prior curriculum-learning literature, where dynamically reordering or reweighting training data has often failed to yield consistent overall gains, we find that a dynamic, stage-based multilingual curriculum *does* meaningfully improve causal language model performance in this data-constrained multilingual setting—though, notably, the same curriculum does not transfer as effectively to masked language models, as we discuss later in this paper. Our main contributions are summarized below:

- **Curriculum-based Multilingual Data Mixing:** We propose a three-stage curriculum learning strategy for multilingual language modeling and show empirically that it improves causal language model (GPT-2) performance over static and naive mixing baselines. Combined with a multilingual data cleaning and multi-level quality-gating pipeline, our approach dynamically adjusts language mixing ratios while preserving informative mixed-language documents to facilitate cross-lingual transfer under a fixed data budget.
- **Custom Vocabulary:** We design a custom BPE tokenizer that balances the vocabulary size across the three target languages. This approach prevents Chinese characters from disproportionately dominating the vocabulary capacity, which saves a significant number of parameters for the actual transformer layers.
- **Comprehensive ablation study:** We conduct a comprehensive ablation study of vocabulary size, context length, hyperparameters, and language model architectures under the BabyLM multilingual constraints.

2 Background and Related Work

2.1 Multilingual Data Mixing

When training a multilingual model, a critical question is how to mix data from different languages. Simple approaches involve either equal mixing or sampling based strictly on original dataset sizes; however, these static methods are suboptimal. As

demonstrated by [Lample and Conneau \(2019\)](#), sampling based strictly on dataset size biases the model heavily towards high-resource languages, causing it to ignore low-resource ones.

To fix this, researchers often use temperature-based sampling to balance high-resource and low-resource languages. But, keeping this mixing ratio static throughout the entire training process ignores how the model’s learning needs change over time. Recent studies suggest that dynamic data mixing, inspired by curriculum learning, yields much better results. For instance, [Zhang et al. \(2021\)](#) introduced a curriculum framework that dynamically schedules the introduction of languages based on the model’s learning state. To address this in our work, we implement a dynamic, three-stage curriculum learning strategy that adjusts language mixing ratios as training progresses, allowing the model to establish high-resource linguistic foundations before tackling harder cross-lingual alignments.

2.2 Tokenization

Multilingual Tokenizer Variations Tokenization has a massive impact on model performance, especially when applying multilingual language models to monolingual tasks ([Rust et al., 2021](#)). Due to the ‘curse of multilinguality’ ([Conneau et al., 2020](#)), a standard shared tokenizer often lacks the vocabulary capacity to represent all languages equitably, underrepresenting certain languages even with pretraining oversampling ([Conneau et al., 2020](#)). If a tokenizer is not carefully adapted, it can aggressively over-segment morphologically rich languages—such as Arabic, Finnish, Korean, or Turkish—leading to high subword fertility that severely hurts downstream performance ([Rust et al., 2021](#)).

Vocabulary Size Constraints Another major issue is the size of the tokenizer’s vocabulary. To support subwords in the Latin alphabet (for English and Dutch) and thousands of characters (for Chinese), the vocabulary must be large. However, in small-scale models typically used in the BabyLM track, a large vocabulary creates a severe parameter bottleneck ([Dagan et al., 2024](#)). Every token in the vocabulary requires an embedding vector. If the vocabulary size is excessively large, the embedding matrix accounts for a substantial proportion of the model parameters. Whether it is more advantageous to allocate parameters to the Transformer backbone rather than to the embedding layer is

ultimately an empirical trade-off: an embedding layer that is too small under-represents the script diversity of the target languages, while one that is too large starves the Transformer backbone of capacity. Our own ablation results (Section 5.1) support this view, showing that performance peaks at an intermediate vocabulary size (32k) rather than improving monotonically as the embedding layer shrinks. Selecting an appropriate vocabulary size therefore constitutes an important architectural design trade-off for efficient multilingual language modeling, rather than a simple rule that smaller embeddings are always better.

2.3 Architectures

Under the limited training budgets of the BabyLM Challenge, model architecture plays an important role in determining data efficiency and downstream performance. We consider both encoder-only Masked Language Models (MLMs), such as BERT (Devlin et al., 2019), and decoder-only Causal Language Models (CLMs), such as GPT-2 (Radford et al., 2019). Notably, encoder-only architectures have proven highly competitive in prior data-efficient settings: in the 2023 BabyLM Challenge, MLM-based submissions such as LTG-BERT achieved the highest average performance, outperforming models trained on vastly larger corpora (Warstadt et al., 2023). GPT-2 additionally serves as the official baseline architecture in the 2026 multilingual track. In this work, we investigate both BERT-style MLMs and GPT-2-style CLMs under different vocabulary sizes, packing strategies, and data mixing strategies to better understand their behavior in multilingual low-resource language modeling.

3 Method

3.1 Data Cleaning and Filtering

To maximize the information density of each language’s training signal within our fixed 100M-token budget—removing noisy or duplicated text that would otherwise consume tokens without contributing to the model’s linguistic competence—we created a step-by-step cleaning process for the English (eng), Dutch (nld), and Chinese (zho) text. Instead of treating all languages the same, we carefully remove low-quality data while keeping the useful parts using three main rules.

First, **Language-Specific Normalization** repairs nonstandard formatting. For example, we

clean up Chinese conversations by standardizing speaker names and adding line breaks. We also remove extra spaces and replace difficult math equations with a simple [formula] tag so the model would not focus on math details during training.

Second, **Redundancy Elimination** removes duplicate text. We convert all characters to lowercase and strip all whitespace and punctuation before checking for duplicates. This ensures the removal of repeated sentences even in the presence of slight typographical variations. Through this process, we successfully filtered out 12 duplicate instances (0.87%) from the English dataset, 654 instances (7.93%) from the Dutch dataset, and 896 instances (5.09%) from the Chinese dataset.

Third, **Multi-Level Quality Gating** applies strict thresholds to filter out low-quality text. We enforce length filters, requiring Chinese documents to be at least 3 characters and English/Dutch documents to be at least 30 characters. Additionally, we implement a stopwords-based purity check specifically for the Dutch corpus to filter out English contamination (the English and Chinese corpora naturally maintain high purity due to source reliability and distinct scripts). Finally, we actively extract mixed-language documents by detecting Latin characters within the Chinese training corpus. We isolate these specific texts to facilitate cross-lingual transfer during Stage 2 of training. The exact rules and formulas for this cleaning process are detailed in Appendix A.

Before training, we randomly take out 5% of the clean data from each language to use as a validation set. The rest of the data is used for training.

3.2 Data Mixing Strategy

The BabyLM competition sets strict limits on training data. Instead of relying on a static data mixing ratio, we propose a dynamic, three-stage curriculum learning strategy strictly regulated by precise Byte-Pair Encoding (BPE) token accounting. All samples in these stages are drawn from the global training pool.

Byte-Premium Sampling Different languages exhibit varying levels of tokenization efficiency, requiring different amounts of token space to convey the same information. This disparity is formally known as the “byte premium” (Arnett et al., 2024). To ensure the actual training tokens for each language strictly adhere to our budget ratios, we implement an exact BPE token calculation mecha-

nism. For a given target budget, our system counts the exact number of processed tokens rather than estimating by character or word count. Furthermore, we introduced premium factors (English is 1.0, Dutch is 1.0516, and Chinese is 0.93) to adjust the allowed sampling volume. We divide our data budget by these numbers. This step ensures we stop training exactly when we hit our data limit.

Baseline Mixing Strategies To strictly evaluate the effectiveness of our proposed curriculum, we implement two comparative baseline strategies:

- **Naive Mixing:** This approach aggregates the raw corpora uniformly without enforcing specific language balancing or strict per-language token budget constraints. It serves as the foundational lower-bound reference for our multilingual modeling.
- **Static Mixing:** This strategy maintains a fixed, predetermined language distribution across the entire training budget. Unlike the dynamic curriculum approach, the sampling ratio (44% English, 23% Dutch, and 33% Chinese) remains completely constant from the initial to the final training step.

Three-Stage Curriculum Mixing We divide the total training budget into three progressive stages, dynamically adjusting the language ratios based on the model’s learning requirements:

- **Stage 1: Foundation (30% of total budget)** This stage focuses on establishing fundamental linguistic structures. We allocate 50% of the stage budget to English, since English benefits from the largest share of our asymmetric BPE tokenizer’s training corpus and therefore the most efficient subword segmentation, allowing the model to quickly build core syntactic and semantic representations before addressing typologically more distant languages. Chinese and Dutch each receive 25%.
- **Stage 2: Alignment (30% of total budget)** Aiming to strengthen cross-lingual representations, we balance the distribution (33% English, 33% Chinese, 34% Dutch). Crucially, we enable an **Aligned Upsampling** mechanism for the Chinese split: half of the Chinese data is sampled from the mixed-language texts identified during preprocessing (Section 3.1), which act as a bridge between the languages

and help the model relate the foundational English representations from Stage 1 to Chinese and Dutch.

- **Stage 3: Hard Boosting (40% of total budget)** Once a foundational cross-lingual mapping is established, we shift the focus toward Chinese and Dutch, whose subword vocabularies are comparatively less optimized due to their smaller share of the tokenizer training corpus, and whose typological distance from English—a character-based script for Chinese, distinct morphosyntax for Dutch—makes cross-lingual transfer more challenging. The distribution is skewed heavily towards Chinese (40%) and Dutch (40%), reducing English to 20%, thereby pushing the model to strengthen its representations of the non-English languages once a shared foundation is in place.

During data mixing at each stage, all samples are globally shuffled. To prevent the model from encountering identical data sequences across different stages, we dynamically generate unique random seeds based on the hash of the stage name. This approach ensures a robust and continuously varied training distribution throughout the curriculum.

3.3 Custom BPE Tokenizer and Asymmetric Sampling

We train our own BPE tokenizer from scratch on the sampled dataset, rather than adopting a massive, ready-made tokenizer. This approach allowed us to use more parameters for the main model by keeping the vocabulary size small. For the 100M budget, we evaluated the vocabulary with 8k, 16k, 32k and 64k. Additionally, to ensure proper processing of Chinese semantics, we explicitly register common Chinese punctuation marks as special tokens.

Asymmetric Sampling Strategy When constructing the tokenizer training corpus, we do not sample the three languages uniformly. Instead, we adopt an asymmetric sampling ratio. For the 100M track, the tokenizer is trained on a 15M-token corpus consisting of 80% English, 17% Dutch, and 3% Chinese.

This distribution is chosen because Chinese characters typically encode more information per character and require substantially larger vocabulary coverage than alphabetic languages. An equal sampling ratio would therefore allocate a disproportion-

ate fraction of the vocabulary to Chinese, reducing the capacity available for English and Dutch subwords. This strategy encourages efficient subword learning for English and Dutch while preserving sufficient coverage of commonly used Chinese characters.

Bootstrapping Workflow The tokenizer training process is integrated into our precise budget-calculation pipeline through a three-step bootstrapping workflow. First, we extract a heuristic dataset from our pre-cleaned global training pool using basic length estimations (character counts for Chinese and whitespace splits for English and Dutch). Second, we train the custom BPE tokenizer on this proxy dataset. Finally, as detailed in Section 3.2, we feed the trained tokenizer back into the main data pipeline to execute exact, BPE-based token counting (adjusted via the Byte Premium). This bootstrapping process guaranteed that our final dataset perfectly adhered to the strict competition budget limits.

3.4 Model Architecture

In this study, we implement and compare two distinct transformer-based architectures.

Autoregressive Baseline (GPT-2): We implement a standard decoder-only GPT-2 architecture. This model relies on causal language modeling (CLM) and we implement two packing strategies: *wrapped-packing* and *bfd-split*.

Bidirectional Encoder (BERT) with Chunking and Wrapping: We implement a BERT-style Masked Language Model (MLM) to learn bidirectional contextual representations and compare three sequence formatting strategies:

- *Naive Chunking:* Documents are truncated or padded to a fixed sequence length (e.g., 32, 128, 256, or 512 tokens). This approach is simple but inherently inefficient, as it either splits semantic context across chunk boundaries or wastes the strict data budget on useless padding tokens.
- *Document Wrapping (Packing):* Multiple documents are concatenated into fixed-length sequences separated by EOS/SEP tokens. This strategy improves sequence utilization by reducing padding overhead while allowing more training tokens to be processed within the fixed token budget.

- *BFD-Split (Best-Fit Decreasing Split):* As an alternative to naive chunking, documents are first sorted by length and then packed into fixed-length sequences using a best-fit decreasing heuristic without crossing document boundaries.

Controlled Token Exposure per Update To enable a fair comparison across different sequence lengths, we control the effective number of tokens processed per optimizer update:

$$E = B \times L \times G,$$

where B denotes the micro-batch size, L the sequence length, and G the gradient accumulation factor.

For the controlled curriculum-learning experiments, we fix $E = 2048$ across sequence lengths of 32, 64, 128, 256, and 512 by adjusting G accordingly. This reduces confounding effects caused by different optimizer update frequencies and enables a more controlled evaluation of sequence-length effects.

4 Experimental Setup

4.1 Dataset

The training data is sampled from the official BabyLM Challenge dataset (BabyLM Multilingual datasets, 2026), built on the BabyBabelLM multilingual corpus (Jumelet et al., 2026a), specifically targeting English (eng), Dutch (nld), and Chinese (zho). To ensure high-quality inputs, we applied a strict data preprocessing pipeline, as detailed in Section 3.1. Furthermore, to account for the information density differences across languages, we implement a Byte Premium scaling factor to adjust the exact token budgets during sampling. For data mixing, we explored strategies including Naive Baseline (equal ratios), Static Baseline (44% EN, 33% ZH, 23% NL), and a curriculum-based progression.

4.2 Tokenizer Settings

We trained a custom BPE tokenizer from scratch using a heuristically sampled multi-lingual corpus (80% English, 17% Dutch, 3% Chinese). The tokenizer is configured with a vocabulary size of 32,000 to fit the 100M track budget. The special tokens include `<unk>`, `<s>`, `</s>`, `<pad>`, `<mask>`, alongside a set of common Chinese punctuation marks to prevent unnecessary fragmentation. A

post-processor is applied to append `<s>` and `</s>` to sequence boundaries.

4.3 Model Configuration

Although we train and evaluate both the MLM and CLM objectives introduced in Section 2.3, we focus our main-text discussion and hyperparameter details on the decoder-only GPT-2 architecture, since, as we show in Section 5.1, GPT-2 consistently outperforms our BERT models across all ablations and official evaluations. Full BERT configurations and results are reported in Appendix C. Table 1 details the GPT-2 configuration used for our 100M track experiments.

Hyperparameter	Value
Model Architecture	GPT-2
Hidden Size	768
Num Layers	12
Attention Heads	12
Inner Dimension	3072
Vocab Size	32,000
Max Seq Length	512

Table 1: Model architecture and training hyperparameters for the GPT-2 baseline.

4.4 Evaluation

We evaluate our models locally using the official BabyLM Evaluation Pipeline (Choshen et al., 2026a), which relies on the Language Model Evaluation Harness (lm-eval). Our zero-shot suite covers BLiMP (Warstadt et al., 2020), MultiBLiMP (Jumelet et al., 2026b), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2019), XStoryCloze (Lin et al., 2022b), and XCOMPS (He et al., 2025); our fine-tuning suite covers ARC (Clark et al., 2018), Belebele (Bandarkar et al., 2024), BMLAMA (Qi et al., 2023), INCLUDE (Romanou et al., 2024), MNLI (Williams et al., 2018), SIB-200 (Adelani et al., 2024), TruthfulQA (Lin et al., 2022a), and XNLI (Conneau et al., 2018). Zero-shot and fine-tuned evaluations are conducted on the three primary languages: eng, nld, and zho. The final overall scores are calculated strictly according to the methodology defined by the official 2026 BabyLM Leaderboard (BabyLM-community, 2026).

5 Results and Analysis

5.1 Ablation Studies

To optimize performance under the strict 100M token limit, we conduct systematic ablation studies focusing on tokenizer vocabulary size, data mixing strategies, sequence packing methods, sequence length, and hyperparameters. Exhaustively testing every combination of these factors was computationally infeasible under our budget. Instead, within each ablation group, we vary one primary factor while holding the remaining settings fixed. Due to computational constraints, different ablation groups were conducted at different model scales: the vocabulary-size and tokenizer experiments use the Standard configuration (hidden size 768, 12 layers, and 12 attention heads), whereas the data-mixing and sequence-length experiments use the Small configuration (hidden size 256, 4 layers, and 4 attention heads). Unless otherwise stated, the experiments use our custom asymmetric BPE tokenizer (32k vocabulary, 80C).

A clear and consistent trend across all experiments is that GPT-2 consistently outperforms the official baseline (38.86), whereas BERT consistently underperforms the baseline across all tested variations.

First, we evaluate different BPE vocabulary sizes—specifically 8,000, 16,000, 32,000, and 64,000 tokens—trained via our custom BPE trainer under our asymmetric sampling strategy (Section 3.3), while keeping the curriculum data mixing, Pack-Wrap packing, and sequence length fixed at their default values. As shown in Table 2, the optimal vocabulary size depends on the model architecture. For GPT-2, a vocabulary size of 32,000 yields the highest overall score (44.06) and the best fine-tuned average across languages (37.32%). Detailed results for BERT are provided in Appendix C, Table 8.

Vocab	Zero-shot (%)				Fine-tune (%)				Score
	en	nl	zh	avg	en	nl	zh	avg	
Base.	56.34	57.81	51.34	55.16	28.34	24.78	27.23	26.78	38.86
8k	52.69	55.51	49.28	52.49	35.87	33.70	34.61	34.73	42.30
16k	53.41	55.84	49.77	53.01	36.16	35.84	34.12	35.37	42.85
32k	53.51	55.88	49.87	53.09	38.81	35.66	37.47	37.32	44.06
64k	53.41	56.38	50.08	53.29	35.57	33.98	37.43	35.66	43.20

Table 2: Ablation results for the Standard GPT-2 configuration (768/12/12) across different BPE vocabulary sizes. Bold values indicate the best performance among the evaluated configurations in each column.

Second, we investigate three distinct data mixing strategies to efficiently allocate the 100M token budget: a *naive* uniform sampling, a *static* weighted sampling based on task relevance, and a *curriculum* learning approach, all using our custom asymmetric BPE tokenizer with the best-performing 32k vocabulary identified above, and default packing and sequence length settings. As shown in Table 3, for GPT-2, the *curriculum* approach achieves the highest Overall Score (41.59) and the best fine-tuned average performance. Detailed results for BERT are provided in Appendix C, Table 9.

Strategy	Zero-shot (%)				Fine-tune (%)				Score
	en	nl	zh	avg	en	nl	zh	avg	
Base.	56.34	57.81	51.34	55.16	28.34	24.78	27.23	26.78	38.86
Naive	52.27	54.19	48.78	51.74	33.91	33.00	33.75	33.56	41.29
Static	52.44	53.95	48.57	51.65	33.87	32.17	34.01	33.35	41.15
Curr.	51.65	54.22	48.85	51.57	35.46	32.02	34.96	34.15	41.59

Table 3: Ablation results for GPT-2 data mixing strategies using the Small configuration (256/4/4). Bold values indicate the best performance among the evaluated strategies in each column.

Furthermore, to validate the effectiveness of our asymmetric sampling strategy, we compared our custom BPE tokenizer against a standard default BPE tokenizer trained on a uniformly sampled multilingual corpus. As demonstrated in Table 4, the custom asymmetric BPE tokenizer yields superior overall performance compared to the standard approach. This confirms that deliberately balancing vocabulary allocation based on the varying information densities of the target scripts is crucial for maximizing tokenization efficiency under strict parameter constraints.

Method	Zero-shot (%)				Fine-tune (%)				Score
	en	nl	zh	avg	en	nl	zh	avg	
Default BPE	53.26	56.68	50.24	53.39	36.05	33.14	34.86	34.68	42.67
Custom BPE	53.51	55.88	49.87	53.09	38.81	35.66	37.47	37.32	44.06

Table 4: Comparison between default uniformly sampled BPE and custom asymmetric BPE using the Standard GPT-2 configuration (768/12/12) with a 32k vocabulary.

Third, we compare sequence packing strategies under the default sequence length of 256 and our best-performing 32k custom BPE tokenizer with curriculum mixing. For GPT-2, we compare Pack-Wrap (document wrapping) against Bfd-Split; as shown in Table 5, Pack-Wrap achieves a marginally

higher overall score (44.55) than Bfd-Split (44.35), indicating that simple document-wrapping concatenation is at least as effective as, and slightly better than, sorting-based best-fit packing for causal language modeling. For BERT, we additionally compare against Naive Chunking; as shown in Table 12 (Appendix C), the three strategies perform comparably (35.42–35.45 overall score), with no packing method showing a consistent advantage for the masked language modeling objective—unlike for GPT-2, where document wrapping shows a small but consistent edge.

Lastly, we evaluate the impact of sequence length and hyperparameter scaling on model performance. The detailed discussions and corresponding results are provided in Appendix Tables 10 and 11.

5.2 Training Dynamics

To evaluate learning stability and capability during the curriculum learning process, we analyze the validation loss progression of GPT-2 and BERT across three distinct training stages. As illustrated in Figure 1, the training dynamics of the two models exhibit significant differences.

For GPT-2, the validation loss decreases consistently across all three curriculum stages, dropping from 4.002 in Stage 1 to 3.498 in Stage 3. This indicates that GPT-2 benefits from the progressively more challenging curriculum.

In contrast, BERT maintains a much higher validation loss and shows only limited improvement during training. For the Pack Wrapped setting, the validation loss changes from 7.612 (Stage 1) to 7.533 (Stage 2) and then increases to 7.640 (Stage 3). The Chunk configuration exhibits a similar trend, with the loss increasing from 7.600 in Stage 1 to 7.820 in Stage 3. Overall, unlike GPT-2, BERT does not achieve continuous improvement throughout the curriculum.

We note two caveats when interpreting these results. First, because MLM and CLM losses are computed over different prediction objectives (bidirectional cloze reconstruction versus unidirectional next-token prediction), their absolute magnitudes are not directly comparable; the loss gap between GPT-2 and BERT in Figure 1 should not be read as evidence that one model is learning more effectively than the other overall, only that their loss scales differ. Second, BERT’s lack of consistent improvement across curriculum stages may not solely reflect an inherent incompatibility between the MLM objective and curriculum-based data mix-

ing; it could also stem from hyperparameters (e.g., learning rate) that were primarily tuned for GPT-2. We leave a dedicated hyperparameter search for BERT under the curriculum setting to future work.

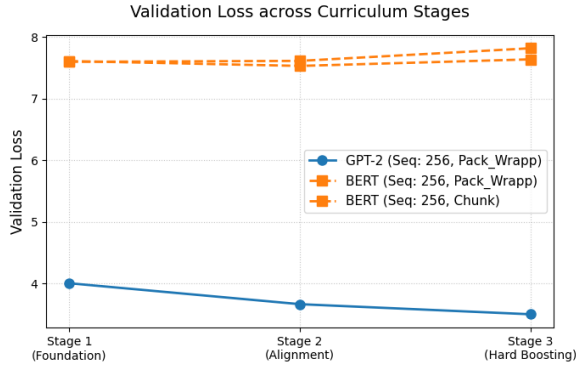


Figure 1: Overview of the validation loss progression of GPT-2 and BERT.

5.3 Official Evaluation Results

We evaluate our models using the official babylm-eval pipeline. Table 5 summarizes the zero-shot and fine-tuned performance of our causal (GPT-2) models, each trained with our best-performing configuration from the ablation studies above (32k custom asymmetric BPE tokenizer, three-stage curriculum mixing, sequence length 256), comparing the two sequence packing variants (Pack-Wrap and Bfd-Split) discussed in Section 5.1. Overall, the GPT-2 models achieve strong results, with the best model reaching an overall score of 44.55, significantly outperforming the official baseline (38.86). While we also evaluated masked (BERT) architectures, the GPT-2 models consistently outperformed their BERT counterparts across our evaluations (detailed BERT results are provided in Appendix 12).

As shown in Tables 6 and 7, our modeling strategies and custom BPE tokenization prove particularly effective for English sib200 fine-tuning, where our curriculum GPT-2 Pack-Wrapped model scores 78.50, outperforming the official baseline of 21.50 by 57.00 points. However, performance on certain zero-shot grammar evaluations, such as English multiblimp, indicates room for improvement; our GPT-2 model scores 79.61, failing to surpass the baseline’s 87.14.

Seq Method	Fine-tune				Zero-shot				Score
	en	zh	nl	avg	en	zh	nl	avg	
- Baseline	28.34	27.23	24.78	26.78	56.34	51.34	57.81	55.16	38.86
256 Pack-Wrap	40.64	36.70	36.79	38.04	53.94	49.78	56.19	53.30	44.55
256 Bfd-Split	40.05	39.05	34.16	37.75	53.34	49.78	56.12	53.08	44.35

Table 5: Official evaluation results for curriculum GPT-2 using the Standard configuration (768/12/12). Bold values indicate the best score among our two GPT-2 packing variants (Pack-Wrap, Bfd-Split) in each column; note that the official baseline outperforms both variants on every zero-shot metric, while our models substantially outperform it on fine-tuning metrics and overall score.

Lang.	Task	Baseline	GPT-2 Pack-Wrap
ENG	blimp	69.18	65.51
	hellaswag	26.66	26.03
	multiblimp	87.14	79.61
	winogrande	48.80	50.04
	xstorycloze	49.92	48.53
	avg	56.34	53.94
NLD	blimp_nl	79.27	75.27
	hellaswag	26.37	26.00
	multiblimp	91.85	88.20
	winogrande	49.30	47.40
	xcomps_nl	52.68	51.59
	xstorycloze	47.37	48.68
	avg	57.81	56.19
ZHO	blimp	75.44	71.86
	hellaswag	27.05	26.75
	winogrande	49.71	49.88
	xcomps_zh	53.90	52.64
	xstorycloze	50.62	47.76
	avg	51.34	49.78
Overall Average		55.16	53.30

Table 6: Zero-shot tasks results for GPT-2.

Lang.	Task	Baseline	GPT-2 Pack-Wrap
ENG	arc	24.79	27.29
	belebele	22.16	25.57
	bmlama	9.35	22.27
	mnli	51.63	50.68
	sib200	21.50	78.50
	truthfulqa	22.32	33.93
	xnli	46.65	46.25
	avg	28.34	40.64
NLD	arc	24.38	26.25
	belebele	30.11	31.82
	bmlama	10.68	23.84
	include	19.64	27.68
	mnli	43.92	49.89
	sib200	21.50	69.50
	truthfulqa	23.21	28.57
	avg	24.78	36.79
ZHO	arc	25.00	24.38
	belebele	22.73	19.89
	bmlama	9.35	19.78
	include	26.79	27.68
	mnli	45.83	54.22
	sib200	21.50	76.00
	truthfulqa	23.21	23.21
	xnli	43.45	48.45
	avg	27.23	36.70
Overall Average		26.78	38.04

Table 7: Fine-tuning task results for GPT-2.

Conclusion

In this work, we present an optimized approach for training small-scale multilingual language models under the strict 100M token of the BabyLM 2026 Multilingual Track. By implementing a custom BPE tokenizer with asymmetric sampling, we effectively balance the vocabulary allocation across languages with distinct information densities, ensuring optimal tokenization efficiency for each script. Furthermore, our dynamic three-stage curriculum learning strategy successfully guides the model from high-resource linguistic foundations to harder, cross-lingual alignments. Our evaluations demonstrate that the GPT-2 architecture adapts exceptionally well to this curriculum, achieving a strong overall score of 44.55 and significantly outperforming the baseline. Coupled with our Pack-Wrapped data packing method, these strategies provide a highly data-efficient framework for multilingual language modeling.

Limitations

Despite the strong overall performance, our approach has several limitations. First, while our asymmetric tokenization strategy aims to balance vocabulary allocation efficiently, heavily skewing the tokenizer corpus towards English and Dutch may introduce slight language-specific biases, potentially limiting the representational depth of Chinese characters. Second, our evaluation reveals that the models struggle to surpass the official baseline in certain zero-shot grammar tasks, such as English `multiblimp`, indicating that the current curriculum may prioritize semantic alignment over strict syntactic rule acquisition. Third, our curriculum stage transitions are highly beneficial for the autoregressive GPT-2 model but fail to yield similar improvements for the masked BERT architecture, suggesting that MLM objectives might require a fundamentally different data mixing schedule. Fourth, strict computational constraints and the 10-epoch limit restrict our capacity for exhaustive hyperparameter tuning and finer-grained stage adjustments. Finally, our ablation studies (Section 5.1) vary one factor at a time while holding the others fixed at a single default configuration, rather than exhaustively testing every combination; the reported optimal settings (e.g., the 32k vocabulary size, the asymmetric sampling ratio) are therefore only shown to be optimal under that particular default configuration, and could interact with other factors in ways we have

not measured.

Another limitation concerns checkpoint reproducibility. Due to storage constraints, the exact checkpoint corresponding to the best exploratory evaluation run was not preserved. The leaderboard submission was therefore produced by retraining the same configuration from scratch. As a result, the leaderboard score may differ from the score reported for the original run. We report the original evaluation results to analyze the effectiveness of the configuration, while the submitted checkpoint should be interpreted as a reproduced model trained under the same setting rather than the identical checkpoint.

Code Availability

The complete source code, preprocessing scripts, tokenization pipelines, and training configurations for this project are publicly available at:

https://anonymous.4open.science/r/BabyLM_Challenge_III-BFCE

Acknowledgments

We are also deeply grateful to the Technical University of Munich (TUM) for providing the essential computational resources through the Leibniz Supercomputing Centre (LRZ) cluster. Furthermore, we would like to thank the BabyLM challenge and Hugging Face for making their datasets publicly available, which were instrumental to the completion of this research.

The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project b279bb. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) – 440719683.

This work was co-funded by the European Union (ERC, EPICAL, 101141712). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Hao-

- nan Gao, and En-Shiun Annie Lee. 2024. [SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, St. Julian’s, Malta.
- Catherine Arnett, Tyler A. Chang, and Benjamin K. Bergen. 2024. A bit of a problem: Measurement disparities in dataset sizes across languages. *arXiv preprint arXiv:2403.00686*.
- BabyLM-community. 2026. BabyLM leaderboard 2026. <https://huggingface.co/spaces/BabyLM-community/BabyLM-Leaderboard-2026>. Accessed: 2026-07-02.
- BabyLM Multi-lingual datasets. 2026. BabyLM 2026 dataset collection. <https://huggingface.co/collections/BabyLM-community/babylm-2026>. Accessed: 2026-06-09.
- Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabisa. 2024. [The Belebele benchmark: a parallel reading comprehension dataset in 122 language variants](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.
- Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Jaap Jumelet, Tal Linzen, Aaron Mueller, Suchir Salhan, Raj Sanjay Shah, Alex Warstadt, and Ethan Gotlieb Wilcox. 2026a. [BabyLM turns 4: Call for papers for the 2026 babylm workshop](#). *arXiv preprint arXiv:2602.20092*.
- Leshem Choshen, Tal Linzen, Raj Sanjay Shah, BabyLM Team, Ryan Cotterell, Mustafa Omer Gul, Aaron Mueller, Alex Warstadt, Jaap Jumelet, Suchir Salhan, and Ethan Gotlieb Wilcox. 2026b. [BabyLM turns 4 and goes multilingual: Call for papers for the 2026 babylm workshop](#). *arXiv preprint arXiv:2602.20092*, v2.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. [Think you have solved question answering? try ARC, the AI2 reasoning challenge](#). *arXiv preprint arXiv:1803.05457*.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.
- Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. [XNLI: Evaluating cross-lingual sentence representations](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium.
- Gautier Dagan, Gabriel Synnaeve, and Baptiste Rozière. 2024. [Getting the most out of your tokenizer for pre-training and domain adaptation](#). In *arXiv preprint arXiv:2402.01035*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Linyang He, Ercong Nie, Sukru Samet Dindar, Arsalan Firoozi, Adrian Florea, Van Nguyen, Corentin Puffay, Riki Shimizu, Haotian Ye, Jonathan Brennan, Helmut Schmid, Hinrich Schütze, and Nima Mesgarani. 2025. [XCOMPS: A multilingual benchmark of conceptual minimal pairs](#). In *Proceedings of the 7th Workshop on Research in Computational Linguistic Typology and Multilingual NLP (SIGTYP 2025)*, Vienna, Austria.
- Jaap Jumelet, Abdellah Fourtassi, Akari Haga, Bastian Bunzeck, Bhargav Shandilya, Diana Galvan-Sosa, Faiz Ghifari Haznitrana, Francesca Padovani, Francois Meyer, Hai Hu, Julien Etxaniz, Laurent Prevot, Linyang He, María Grandury, Mila Marcheva, Negar Foroutan, Nikitas Theodoropoulos, Pouya Sadeghi, Siyuan Song, and 7 others. 2026a. [BabyBabelLM: A multilingual benchmark of developmentally plausible training data](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, Rabat, Morocco.
- Jaap Jumelet, Leonie Weissweiler, Joakim Nivre, and Arianna Bisazza. 2026b. [MultiBLiMP 1.0: A massively multilingual benchmark of linguistic minimal pairs](#). *Transactions of the Association for Computational Linguistics*, 14.
- Guillaume Lample and Alexis Conneau. 2019. [Cross-lingual language model pretraining](#). In *Advances in Neural Information Processing Systems*, volume 32.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022a. [TruthfulQA: Measuring how models mimic human falsehoods](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland.
- Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O’Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, and 2 others.

- 2022b. [Few-shot learning with multilingual generative language models](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052. Introduces the XStoryCloze evaluation set.
- Jirui Qi, Raquel Fernández, and Arianna Bisazza. 2023. [Cross-lingual consistency of factual knowledge in multilingual language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10650–10666, Singapore.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). Technical Report 1, OpenAI.
- Angelika Romanou, Negar Foroutan, Anna Sotnikova, Zeming Chen, Sree Harsha Nelaturu, Shivalika Singh, Rishabh Maheshwary, Micol Altomare, Mohamed A. Haggag, Snegha A, Alfonso Amayuelas, Azril Hafizi Amirudin, Viraat Aryabumi, Danylo Boiko, Michael Chang, Jenny Chim, Gal Cohen, Aditya Kumar Dalmia, Abraham Diress, and 40 others. 2024. [INCLUDE: Evaluating multilingual language understanding with regional knowledge](#). In *arXiv preprint arXiv:2411.19799*.
- Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. [How good is your tokenizer? on the monolingual performance of multilingual language 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)*, pages 3118–3135.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. [WinoGrande: An adversarial winograd schema challenge at scale](#). *arXiv preprint arXiv:1907.10641*.
- Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjape, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. [Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora](#). In *Proceedings of the 27th Conference on Computational Natural Language Learning: Volume 2: The BabyLM Challenge*, pages 1–34, ETH Zürich. Association for Computational Linguistics.
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. [BLiMP: The benchmark of linguistic minimal pairs for english](#). *Transactions of the Association for Computational Linguistics*, 8:377–392.
- Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [HellaSwag: Can a machine really finish your sentence?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy.
- Mingliang Zhang, Fandong Meng, Yunhai Tong, and Jie Zhou. 2021. [Competence-based curriculum learning for multilingual machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2481–2493, Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Full Data Cleaning Rules

This appendix explains the exact rules mentioned in Section 3.1.

A.1 Text Normalization and Changes

- **Cleaning Chinese Conversations:** We change speaker names to standard ones (for example, changing *TEACHER: to lao shi:). We also add line breaks (`\n`) before a new person speaks to help the model understand who is talking.
- **Changing Math Equations:** We categorize math equations using regular expressions. "Simple" formulas, defined strictly as those containing only alphanumeric characters (e.g., single variables like x or y), are kept in plain text by stripping their surrounding LaTeX tags. Conversely, any complex formula containing spaces, operators, or special LaTeX syntax is replaced entirely with a single `[formula]` tag.

A.2 Removing Duplicates and Strict Quality Rules

- **Length Filter:** We throw away text that is too short. Chinese text (zho) must be at least **3 characters** long, and English (eng) and Dutch (nld) must be at least **30 characters** long. We also delete any text that has more than 3 `[formula]` tags.
- **Dutch Purity Check:** To make sure the Dutch text does not have too much English, we count the common words (stopwords). We throw away the text if there are more English common words than Dutch common words. Also, if a Dutch sentence has more than 10 words but no Dutch common words, we delete it.

- **Finding Mixed Languages:** We give special importance to Chinese text that also has some English letters. This shows the text mixes both languages, which helps connect them in Stage 2 of training.

B Training System and Token Exposure Tracking

Models are trained using the Hugging Face Transformers framework and optimized with the AdamW optimizer using a learning rate of 1×10^{-4} .

To ensure compliance with the multilingual BabyLM token budget, we maintain runtime token-exposure tracking during training. Token counts are monitored using the attention mask and language identifiers associated with each training sample. The adjusted multilingual exposure follows the byte-premium weighting scheme described in Section 3.2.

Training is controlled by adjusted token exposure, with a maximum of 10 epochs. Before processing each batch, the trainer estimates the additional adjusted exposure that the batch would introduce. If adding the batch would exceed the current token budget, the batch is skipped and training terminates at the last valid state. This strict pre-check ensures that the final adjusted exposure never exceeds the 100M adjusted-token limit.

Padding tokens are excluded from exposure accounting. For GPT-2 packed training, artificial document-boundary EOS tokens are assigned ignored language identifiers and excluded from language-specific exposure accounting. For curriculum-learning experiments, the same strict exposure tracking is applied cumulatively across stages.

C Detailed Ablation Results and Training Dynamics

Vocab	Zero-shot (%)				Fine-tune (%)				Score
	en	nl	zh	avg	en	nl	zh	avg	
Base.	56.34	57.81	51.34	55.16	28.34	24.78	27.23	26.78	38.86
8k	49.68	49.18	47.52	48.80	25.42	25.80	25.59	25.60	35.38
16k	49.76	49.45	47.40	48.87	25.42	25.80	25.59	25.60	35.42
32k	49.49	49.78	47.54	48.94	25.63	25.17	25.78	25.53	35.42
64k	50.09	50.14	47.02	49.08	25.23	25.80	25.42	25.48	35.45

Table 8: Ablation results for the Standard BERT configuration (768/12/12) across different BPE vocabulary sizes.

Strategy	Zero-shot (%)				Fine-tune (%)				Score
	en	nl	zh	avg	en	nl	zh	avg	
Base.	56.34	57.81	51.34	55.16	28.34	24.78	27.23	26.78	38.86
Naive	46.82	46.33	45.17	46.11	30.70	27.43	32.61	30.25	37.00
Static	47.10	47.41	45.68	46.73	31.53	29.07	33.15	31.25	37.84
Curr.	47.54	45.23	46.57	46.44	32.04	28.27	32.17	30.83	37.43

Table 9: Ablation results for BERT data mixing strategies. Ablation results for BERT data mixing strategies using the Small configuration (256/4/4).

We analyze the impact of sequence length on model performance and training efficiency, evaluating lengths of 32, 64, 128, 256, and 512 tokens. According to Table 10, a sequence length of 512 with GPT-2 achieves the highest Overall Score (41.54). For BERT, performance peaks at an Overall Score of 37.71 with a sequence length of 128.

Finally, we compare hyperparameters, specifically a standard configuration ($h = 768, L = 12, a = 12$) against a smaller configuration ($h = 256, L = 4, a = 4$). As detailed in Table 11, scaling up the model dimensions yields higher overall scores for GPT-2 (44.06). Interestingly, while the standard configuration consistently improves GPT-2’s performance across all metrics, BERT exhibits a trade-off: scaling up shows strong gains in zero-shot capabilities but causes a noticeable decline in its average fine-tuned performance (dropping from 30.25% to 25.53%).

Model	Seq	Zero-shot (%)				Fine-tune (%)				Score
		en	nl	zh	avg	en	nl	zh	avg	
Base.	-	56.34	57.81	51.34	55.16	28.34	24.78	27.23	26.78	38.86
GPT-2	32	50.49	54.04	49.67	51.40	26.31	26.54	32.98	28.61	38.34
	64	51.92	54.48	48.81	51.74	30.94	29.21	34.01	31.38	40.09
	128	52.13	54.63	48.50	51.75	31.41	30.04	35.22	32.22	40.59
	256	52.27	54.19	48.78	51.74	33.91	33.00	33.75	33.56	41.29
	512	51.28	53.81	48.26	51.12	34.22	33.69	35.33	34.41	41.54
BERT	32	45.84	44.29	46.35	45.50	29.55	28.61	30.77	29.64	36.32
	64	46.75	45.19	45.78	45.91	31.79	28.44	32.93	31.05	37.35
	128	46.21	46.29	45.30	45.94	33.61	28.58	32.63	31.61	37.71
	256	46.82	46.33	45.17	46.11	30.70	27.43	32.61	30.25	37.00
	512	48.12	46.42	46.19	46.91	30.11	27.92	32.09	30.04	37.19

Table 10: Sequence-length ablation results using the Small GPT-2 and BERT configurations (256/4/4).

Model Params	Zero-shot (%)				Fine-tune (%)				Score
	en	nl	zh	avg	en	nl	zh	avg	
Base. -	56.34	57.81	51.34	55.16	28.34	24.78	27.23	26.78	38.86
GPT-2 Small	52.27	54.19	48.78	51.74	33.91	33.00	33.75	33.56	41.29
	53.51	55.88	49.87	53.09	38.81	35.66	37.47	37.32	44.06
BERT Small	46.82	46.33	45.17	46.11	30.70	27.43	32.61	30.25	37.00
	49.49	49.78	47.54	48.94	25.63	25.17	25.78	25.53	35.42

Table 11: Ablation results comparing different hyperparameter configurations.

In addition to the GPT-2 models discussed in the main text, we evaluated masked language models using the BERT architecture. Table 12 summarizes the official evaluation results for BERT trained with static data mixing strategies and chunking or BFD-split methods.

The highest overall score achieved by our BERT models is 35.45, which underperforms compared to both the official baseline (38.86) and our GPT-2 models (44.55). Detailed performance breakdowns for BERT across individual zero-shot and fine-tuning tasks are provided in Tables 13 and 14, respectively. Although BERT struggles to match the baseline overall, the BFD-Split strategy with a sequence length of 128 slightly improves its fine-tuning average (25.48%) and zero-shot average (49.11%) compared to naive chunking. The results suggest that the masked language modeling objective, when constrained by strict vocabulary limits and the 100M token budget, may require substantially different hyperparameter tuning or training durations compared to autoregressive models.

Seq Method	Fine-tune				Zero-shot				Score
	en	zh	nl	avg	en	zh	nl	avg	
- Baseline	28.34	27.23	24.78	26.78	56.34	51.34	57.81	55.16	38.86
256 Chunk	25.63	25.78	25.17	25.53	49.78	47.82	49.47	49.02	35.45
256 Pack-Wrap	25.23	25.42	25.80	25.48	49.97	47.54	49.63	49.05	35.42
128 Bfd-Split	25.23	25.42	25.80	25.48	49.74	47.68	49.91	49.11	35.45

Table 12: Official evaluation results for selected Standard BERT configurations (768/12/12). Bold values indicate the best score among our three BERT packing/formatting variants (Chunk, Pack-Wrap, Bfd-Split) in each column; note that the official baseline outperforms all variants on every zero-shot metric.

Lang.	Task	Baseline	BERT Bfd-split
ENG	blimp	69.18	59.89
	hellaswag	26.66	25.98
	multiblimp	87.14	65.84
	winogrande	48.80	50.29
	xstorycloze	49.92	46.67
NLD	blimp_nl	79.27	64.57
	hellaswag	26.37	25.60
	multiblimp	91.85	62.63
	winogrande	49.30	50.29
	xcomps_nl	52.68	50.45
	xstorycloze	47.37	45.90
ZHO	blimp	75.44	62.27
	hellaswag	27.05	26.12
	winogrande	49.71	49.05
	xcomps_zh	53.90	52.14
	xstorycloze	50.62	48.84

Table 13: Zero-shot tasks results for BERT.

Lang.	Task	Baseline	BERT Bfd-split
ENG	arc	24.79	24.79
	belebele	22.16	26.14
	bmlama	9.35	9.35
	mnli	51.63	35.59
	sib200	21.50	21.50
	truthfulqa	22.32	27.68
	xnli	46.65	31.55
	avg	28.34	25.23
NLD	arc	24.38	24.38
	belebele	30.11	26.14
	bmlama	10.68	9.52
	include	19.64	34.82
	mnli	43.92	36.54
	sib200	21.50	21.50
	truthfulqa	23.21	27.68
	avg	24.78	25.80
ZHO	arc	25.00	24.79
	belebele	22.73	26.14
	bmlama	9.35	9.35
	include	26.79	26.79
	mnli	45.83	35.59
	sib200	21.50	21.50
	truthfulqa	23.21	27.68
	xnli	43.45	31.55
	avg	27.23	25.42

Table 14: Fine-tuning task results for BERT.