Qwenfant: Train Big Character-level BabyLMs on Concise Parallel Text

Lukas Edman[1,2] **Alexander Fraser**[1,2,3]

[1]School of Computation, Information and Technology, TU Munich
[2]Munich Center for Machine Learning
[3]Munich Data Science Institute

lukas.edman@tum.de

Abstract

This work concerns our submission for the MULTILINGUAL track of the 2026 BabyLM Challenge.[1] We improve on our baseline using data that has been summarized and translated by an LLM, showing that higher-quality data can come from synthetic sources. We further improve with 2 simple changes: using a larger model, and using a character-level vocabulary. Both of these changes require LoRA-style fine-tuning for realizing their full potential. We dub our final Qwen3-based model Qwenfant, which, at the time of submission, ranks 1st in the MULTILINGUAL track.

1 Introduction

Language Models (LMs) are generally considered to be much less data-efficient than humans, requiring orders of magnitude more data to achieve similar performance. However the nature of the data seen by humans is vastly different from that typically seen by these LMs. The majority of LM training data is mined from the web, with little to no regard to its lexical complexity or even grammaticality.

BabyLM and BabyBabelLM have introduced datasets that include training data that would be more likely for a child to encounter, such as child-directed speech, however participants have found that some of this data degrades performance on standard benchmarks (Padovani et al., 2025).

Optimizing a dataset through selection is a difficult, if not intractable, task. Instead, we propose a simpler method: use an existing pretrained Large Language Model (LLM) to improve the conciseness of the data. Pretrained LLMs have remarkable ability to summarize text, removing unimportant information such as filler words, or overly verbose descriptions. LLMs are also adept at translating between languages, allowing us to generate high-quality synthetic parallel training data. We experiment with both summarizing and translating the BabyLM Challenge's provided English data to Dutch and Chinese, the other two languages involved in the MULTILINGUAL track.

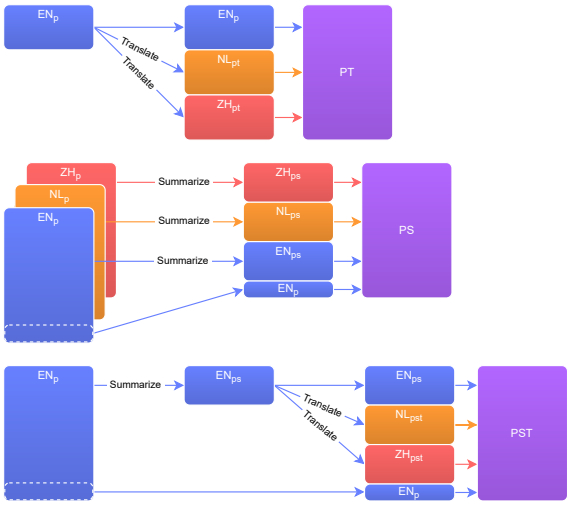

Figure 1: Preprocessing pipelines we investigate, including packed-translated (*PT*), packed-summarized (*PS*), and packed-summarized-translated (*PST*). Packing process not shown for brevity.

In this work, we find that:

- Using 3 straightforward preprocessing steps, packing, summarization, and translation, provides a large performance improvement across all languages.
- Using a larger model is beneficial, providing more capacity for each language, but only when using LoRA for fine-tuning to prevent overfitting.
- Using a character-level tokenization also improves results in larger models, when using LoRA.

[1]https://github.com/Leukas/babylm26

Proceedings of the Second BabyLM Workshop, pages 624–630
October 29, 2026 ©2026 Association for Computational Linguistics

These three improvements culminate in an 11-point increase over the baseline, marking a substantially higher quality multilingual BabyLM.

2 Related Work

2.1 Multilingual Pretraining

Multilingual pretraining is typically approached with a similar strategy as monolingual pretraining: training an LM with masked or causal language modeling, with the only major change being an increased vocabulary size to better accomodate the diverse character sets. This was adopted as the standard since the release of mBERT (Devlin et al., 2019), and has remained effectively the same to this day. One of the latest examples of a performant multilingual model includes Qwen 3 (Yang et al., 2025), which uses this same pretraining strategy, only with some architectural optimizations and adding a reasoning fine-tuning step.

2.2 Synthetic Training Data

A number of works have investigated the use of synthetic pretraining data for training models, both in and outside the BabyLM context. Some BabyLM submissions have tried to generate synthetic data on-the-fly, with limited success (Ulm et al., 2025; AlKhamissi et al., 2024). Theodoropoulos et al. (2024) and Poh et al. (2025) generated synthetic data (bedtime stories and child-directed speech, respectively), finding this data negatively impacted overall performance. Edman et al. (2024a) used a synthetic dataset that includes sentences and contrastive pairs, and showed improved results on GLUE tasks with a similar structure such as MRPC.

Outside of BabyLM, Eldan and Li (2023) created TinyStories, a synthetic dataset mimicking stories for 3-4 year olds, and found that small language models trained on this could generate fluent outputs with a limited vocabulary. Synthetically-generated textbooks have also been successfully used to pretrain a large-scale LLM (Gunasekar et al., 2023). Kang et al. (2025) built upon this and showed that synthetic data alone does not show an improvement in training efficiency, however a mixture of real and synthetic data does.

2.3 Tokenization

Character-level models have been extensively explored, and have been shown to be more sample-efficient in training, despite their higher training cost overall due to longer sequence lengths (Xue et al., 2022). They have been shown to be more performant on tasks requiring character-level manipulation (Pagnoni et al., 2025), and similar-language translation (Edman et al., 2024b).

In the context of BabyLM, character-level models have been seldom explored. Edman and Bylinina (2023) attempted using a character-level tokenizer without much success, and Edman and Fraser (2025) used character-level information to inform an n-hot encoding, which showed improved adjective nominalization ability.

3 Method

We break down our method into the three improvements that we make over the baseline: preprocessing, tokenization, and model size.

3.1 Preprocessing

Our preprocessing involves 3 steps: packing, summarizing, and translation. For the latter two steps, we use `Qwen2.5-3B-Instruct` (Qwen Team, 2024)[2]. We show the prompts used in Figure 2 in Appendix B.

Packing Packing is simply done by concatenating examples together until they form a desired context length. We use a context length of 512 words. This is primarily done because the structure of examples in the datasets provided by BabyLM does not preserve document boundaries, so for most examples, the surrounding examples are from a related context.

Summarization Summarization is done with the purpose of improving the conciseness of the dataset. Some of the data, particularly that which is transcribed speech, has several filler words or phrases (e.g., 'uh', 'you know'), or repetitions that do not add semantic value. While some of these are easily detectable with simple rules, others are more difficult to replace without understanding the context. We therefore use an LLM to shorten the text.

Following the packing step, we feed the packed examples into the model, asking it to shorten the text to 200 words or less. We note that in our prompt we specify to 'shorten' the text, rather than use the word 'summarize'. This is because for chunks of dialogue, a summary may imply a 3rd-person overview of the conversation. We wish to preserve the dialogue format.

[2] `https://hf.co/Qwen/Qwen2.5-3B-Instruct`

Text	Summary
Well it's just that, you know, a pound, or a hundred pounds today, is not the same as a hundred pounds in a year's time, or two, two years' time. Right. So that would be opportunity cost? That's right, yes, that's right, so, in actual fact, this area here, right, is going to be the discounted sum of all future rural incomes.	A pound today isn't the same as a pound next year. This represents opportunity cost. The discounted sum of future rural incomes is represented here.
A: you can smell ether in the air sometimes A: It keeps the neighborhoods happy I guess. B: I guess so. A: Uh, cause they clean the, uh, the lenses for cameras A: and they, they make film here A: but it's, you know, it's under so much control. A: You know, sulphur dioxide is the big emittant from them A: but that's really getting under control now. B: Right. B: I, you know, I don't know in air what they do. B: Uh, I, I haven't run across any major pollutants down there that I've, I've really seen a lot of, you know, about. B: Uh, I, I know I've seen like, uh, B: my grandparents live in Corpus Christi, Texas B: and I know they've, there's a lot of refineries down there B: and that, that's some pretty potent stuff they put up in the air. B: I, but I don't know how, uh, you know, B: there's a difference in what you can smell and what you, uh, you know, what's bad. A: Be interesting to see when, as Mexico develops industrially whether, you know, without	A: You can smell ether in the air sometimes. It keeps the neighborhoods happy. They clean camera lenses and make film. But it's controlled well. Sulphur dioxide is a big emitter, but it's getting under control now. B: I guess so. I don't know much about air pollution. I've never seen many major pollutants around. My grandparents live near Corpus Christi with lots of refineries, which can be potent. But smelling something doesn't mean it's harmful. It would be interesting to see how Mexico develops industrially.

Table 1: Examples of original text and their respective LLM-generated summaries.

We show examples of the summarization in Table 1. The summaries show a complete removal of filler words, resulting in a significant reduction of the text length. The summaries are however not perfect. The dialogue contains less interaction between the persons, and it even mistakenly attributes the last sentence to person B rather than person A. Nevertheless, we do not need the summarized text to be a perfect reflection of the original text, we only need it to be high-quality training data.

Translation Translation is done to create parallel data. Training on parallel data is typical strategy in MT and generally leads to better alignment, but finding high-quality human-translated parallel data is difficult, particularly trilingual data covering English, Dutch, and Chinese. Meanwhile, LLMs can produce high-quality translations. We therefore translate our English data to Dutch and Chinese using the packed examples or the packed-summarized examples.

With these example triplets, we then concatenate each triplet together, using language tags to separate the languages. We show an example below:

```
The sun shines. <dutch> De zon
schijnt. <chinese> 阳光照耀。
```

We vary the order of the languages, using all 6 permutations equally often. The goal of this format is aligned with the auto-regressive nature of a decoder model: The first part teaches the model to predict the next token based only on its knowledge of the first language, while the subsequent parts let the model use its understanding of the previous languages to inform what should come next. This encourages the model to form an internal mapping between languages. In practice, each part of the triplet consists of several sentences (using our summarized context size of 200 English words), so the model is effectively trained on something more akin to paragraph-level translation.

Pipeline Given these 3 preprocessing steps, we assemble them into different pipelines to ablate their effect. The pipelines include packed-only (*P*), packed-translated (*PT*), packed-summarized (*PS*), and packed-summarized-translated (*PST*). We visualize the latter three in Figure 1.

The datasets we use are provided by the organizers: `BabyLM-2026-Strict`[3] for English, `babylm-nld`[4] for Dutch, and `babylm-zho`[5] for Chinese. *P* and *PT* require only one third of the datasets.

For *PS* and *PST*, the summarization process reduces the data to less than 33% of the original size, so we fill the remaining budget (6.2% and 32.3%, respectively) with packed English data.

3.2 Model Size

A natural question is whether the default model size has enough capacity to adequately support

[3] https://hf.co/datasets/BabyLM-community/BabyLM-2026-Strict

[4] https://hf.co/datasets/BabyLM-community/babylm-nld

[5] https://hf.co/datasets/BabyLM-community/babylm-zho

all 3 languages, while simultaneously preventing overfitting. To test this, we increase the hidden dimension size from 768 to 1024, intermediate size from 3072 to 4096, and number of layers from 12 to 24. This results in a roughly 4× larger model.

The main obstacle for training a larger model, aside from additional compute, is an increased risk of overfitting. For the pretraining phase, this is less of an issue, given that the 100M word budget allows for diverse enough text. Finetuning, on the other hand, does become problematic. To handle this, we use LoRA (Hu et al., 2022), which enables us to more conservatively update model parameters for a specific task. We use a LoRA rank of 256, alpha 512, 5% dropout, and learning rate 1e-4, applying LoRA to all linear layers except the gating layer.

3.3 Tokenization

Choosing an optimal vocabulary size is a nontrivial task, particularly for multilingual models, where one must balance tokens across languages. Our preliminary experiments found that overall, a vocabulary size of 20k was most effective. However, upon closer inspection, character-level models perform better on zero-shot tasks, but substantially worse on finetuning tasks, bringing down their overall average.

While out-of-the-box subword models outperform character models, 2 simple fixes reverse this finding. The first fix is identical to our fix for increasing model size: use LoRA. The second fix is similarly simple, increase the default context length used in finetuning from 128 to 512, accounting for the increased sequence length caused by character-level tokenization.

3.4 Experimental Setup

We use the Qwen3 model architecture (Yang et al., 2025) for our experiments. The primary reason for this is that Qwen3 uses RoPE (Su et al., 2021), which provides a useful bias towards local attention, eliminating the need for a scheduled context size increase as done in Edman and Bylinina (2023).

We note the full hyperparameter details in Table 7 in Appendix A. Our evaluation uses the MULTILINGUAL track's zero-shot and fine-tune tasks.

4 Results

4.1 Data Preprocessing

We show the effect of our preprocessing steps in Table 2. The packing step shows the largest positive effect, which makes sense as some of the examples are exceedingly short on their own, such as someone simply saying "Right." The additional context allows a model to learn longer-distance relationships between sentences, which is useful in particular for the downstream tasks requiring finetuning.

Preprocessing	∅	*P*	*PT*	*PS*	*PST*
Zero-shot					
eng	57.8	55.4	55.8	57.2	**59.0**
nld	**59.9**	58.2	54.0	55.6	56.5
zho	51.9	52.2	49.2	52.5	**52.8**
Fine-tune					
eng	29.1	41.0	43.8	40.9	**45.8**
nld	32.3	39.0	39.5	37.4	**40.2**
zho	20.3	36.5	40.5	41.7	**42.5**
avg	39.5	45.8	46.3	46.4	**48.5**

Table 2: Effect of the 3 preprocessing steps: Packing (*P*), Summarization (*S*), and Translation (*T*)

Model size	Base		Large	
LoRA?	No	Yes	No	Yes
Zero-shot				
eng	59.0	59.0	**59.5**	**59.5**
nld	56.5	56.5	**57.8**	**57.8**
zho	52.8	52.8	**52.9**	**52.9**
Fine-tune				
eng	**45.8**	45.5	21.7	45.0
nld	40.2	40.2	21.4	**41.3**
zho	42.5	42.7	20.3	**44.7**
avg	48.5	48.5	36.2	**49.3**

Table 3: Effect of model size and using LoRA for finetuning.

Beyond packing, both translation and summarization on their own provide a positive effect, but their combination yields the most improvement. We suspect that translation on its own may be not as effective as summarization then translation because the data being translated is of lower quality, and this is improved by the summarization step. Meanwhile, the summarization on its own, while improving data quality, does not benefit from data being parallel.

4.2 Model Size

The effect of model size can be seen in Table 3. The larger model has a higher zero-shot performance across all languages, but suffers from overfitting when fully fine-tuned. LoRA resolves this issue, increasing the fine-tuned performance on Dutch and Chinese particularly.

	20k		Char	
LoRA	No	Yes	No	Yes
Zero-shot				
eng	59.0	59.0	**60.9**	**60.9**
nld	56.5	56.5	**58.6**	**58.6**
zho	**52.8**	**52.8**	52.7	52.7
Fine-tune				
eng	**45.8**	45.5	37.0	44.3
nld	40.2	40.2	33.0	**41.0**
zho	42.5	**42.7**	39.8	42.1
avg	48.5	48.5	45.5	**48.8**

Table 4: Comparison of 20k vocabulary to character-level vocabulary, and the effect of LoRA.

4.3 Tokenization

We show the impact of vocabulary size in Table 5. By default, character models outperform a larger vocabulary on zero-shot tasks for English and Dutch, likely due to a better ability to generalize between the two Latin-scripted languages. However, on finetuning tasks, with no changes, we see a drop in performance. Using a larger context size of 512 tokens and LoRA, the performance recovers. Although the fine-tuning results are still slightly lower for English and Chinese, we see later in Table 5 that with a larger model size it outperforms the larger vocabulary for English. As a character-level model has fewer parameters dedicated to the embeddings and prediction layer (roughly 6-7$\times$ fewer), it may be the case that the base character-level model is too small to fully accomodate a character-level representation. More work would be needed to confirm this.

A character-level vocabulary for Chinese is likely less impactful since Chinese characters contain much more information on average, so our models are still effectively operating on a subword level for Chinese. Chinese characters can be decomposed into Kangxi radicals, and this may be a better equivalent for the character-level granularity of English and Dutch. We leave this for future work.

4.4 Overview

We show our progression from the baseline scores to our final model in Table 5. Here, we can see the impact of our methods as a whole. The majority of the performance improvement comes from preprocessing steps, roughly 9 points. Using a larger model and character-level tokenizer only improve the performance marginally. The overall improvement of our model over the organizers' baseline is around 11 points.

Model	*O.B.*	∅	*P*	*PST*	*PST* Large	*PST* Char Large
Zero-shot						
eng	56.4	57.8	55.4	59.0	59.5	**61.4**
nld	57.8	**59.9**	58.2	56.5	57.8	59.1
zho	51.3	51.9	52.2	52.8	**52.9**	**52.9**
Fine-tune						
eng	28.3	29.1	41.1	45.8	45.0	**47.5**
nld	24.8	32.3	39.1	40.2	41.3	**41.4**
zho	27.2	20.3	36.5	42.5	**44.7**	44.4
avg	38.9	39.5	45.8	48.5	49.3	**50.2**

Table 5: Comparison of the official baseline (*O.B.*) compared to various steps added, and final submitted model (right).

Model	*O.B.*	*PST* Char	*PST* Char Large
Zero-shot			
eng	53.6	57.1	**57.4**
nld	55.0	55.3	**56.1**
zho	47.3	49.3	**51.0**
Fine-tune			
eng	43.1	50.4	**53.3**
nld	39.8	47.7	**48.1**
zho	40.7	47.5	**49.6**
avg	45.9	50.9	**52.4**

Table 6: Comparison of the official baseline (*O.B.*) to our two leaderboard submissions, including hidden tasks. *PST Char* and *PST Char Large* are equivalent to *Qwenfant-0.1B* and *Qwenfant-0.4B*, respectively.

Our final model submissions, with averages including the added hidden tasks, is shown in Table 6. We opted to submit our character-level models, one larger 0.4B model for optimal performance, and one smaller 0.1B model, as this is a standard size and roughly equivalent to the official baseline size. We can see that our preprocessing and tokenization strategy bring 5 points of improvement alone, and increasing the model size brings another 1.5 points.

5 Conclusion

When pretraining multilingual models on limited data, we see that we can improve on out-of-the-box methods in a few different ways. The main improvement we see comes from our preprocessing pipeline, where a pack-summarize-translate strategy is particularly effective at both providing higher-quality data and providing parallel data. Models trained on this data perform consistently well, showing that synthetic data can be valuable,

if not better, than natural data. We also found that character-level models and larger models that typically suffer from overfitting can be coupled with a controlled fine-tuning using LoRA, thereby preventing this overfitting.

These methods provide a sample-efficient pretraining strategy that improves performance on English, Dutch, and Chinese. Future work should extend this to further languages, in order to see if these strategies scale to a massively multilingual language model. It also remains to be seen if these methods have diminishing returns in a less sample-efficient setting.

Acknowledgments

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

Badr AlKhamissi, Yingtian Tang, Abdülkadir Gökce, Johannes Mehrer, and Martin Schrimpf. 2024. Dreaming out loud: A self-synthesis approach for training vision-language models with developmentally plausible data. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 244–251, Miami, FL, USA. Association for Computational Linguistics.

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.

Lukas Edman and Lisa Bylinina. 2023. Too much information: Keeping training simple for BabyLMs. pages 89–97, Singapore.

Lukas Edman, Lisa Bylinina, Faeze Ghorbanpour, and Alexander Fraser. 2024a. Are BabyLMs second language learners? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 166–173, Miami, FL, USA. Association for Computational Linguistics.

Lukas Edman and Alexander Fraser. 2025. Mask and you shall receive: Optimizing masked language modeling for pretraining BabyLMs. In *Proceedings of the First BabyLM Workshop*, pages 445–453, Suzhou, China. Association for Computational Linguistics.

Lukas Edman, Gabriele Sarti, Antonio Toral, Gertjan van Noord, and Arianna Bisazza. 2024b. Are character-level translations worth the wait? comparing ByT5 and mT5 for machine translation. *Transactions of the Association for Computational Linguistics*, 12:392–410.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*.

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. 2023. Textbooks are all you need. *Preprint*, arXiv:2306.11644.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. 2024. Muon: An optimizer for hidden layers in neural networks.

Feiyang Kang, Newsha Ardalani, Michael Kuchnik, Youssef Emad, Mostafa Elhoushi, Shubhabrata Sengupta, Shang-Wen Li, Ramya Raghavendra, Ruoxi Jia, and Carole-Jean Wu. 2025. Demystifying synthetic data in LLM pre-training: A systematic study of scaling laws, benefits, and pitfalls. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10739–10758, Suzhou, China. Association for Computational Linguistics.

Francesca Padovani, Bastian Bunzeck, Manar Ali, Omar Momen, Arianna Bisazza, Hendrik Buschmeier, and Sina Zarrieß. 2025. Dialogue is not enough to make a communicative BabyLM (but neither is developmentally inspired reinforcement learning). In *Proceedings of the First BabyLM Workshop*, pages 421–435, Suzhou, China. Association for Computational Linguistics.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret

Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. 2025. Byte latent transformer: Patches scale better than tokens. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258, Vienna, Austria. Association for Computational Linguistics.

Whitney Poh, Michael Tombolini, and Libby Barak. 2025. What did you say? generating child-directed speech questions to train LLMs. In *Proceedings of the First BabyLM Workshop*, pages 237–245, Suzhou, China. Association for Computational Linguistics.

Shikai Qiu, Charlie Chen, Hoang Phan, Qi Lei, and Andrew Wilson. 2026. Hyperparameter transfer enables consistent gains of matrix-preconditioned optimizers across scales. *Advances in Neural Information Processing Systems*, 38:130867–130911.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. 2021. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*.

Nikitas Theodoropoulos, Giorgos Filandrianos, Vassilis Lyberatos, Maria Lymperaiou, and Giorgos Stamou. 2024. BERTtime stories: Investigating the role of synthetic story data in language pre-training. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 308–323, Miami, FL, USA. Association for Computational Linguistics.

Jannek Ulm, Kevin Du, and Vésteinn Snæbjarnarson. 2025. Contrastive decoding for synthetic data generation in low-resource language modeling. In *Proceedings of the First BabyLM Workshop*, pages 29–41, Suzhou, China. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

A Training Hyperparameters

We show the complete hyperparameters used in Table 7. Notably, we use the Muon optimizer (Jordan et al., 2024) along with AdamW for the embedding parameters (as Muon only works for 2D parameters). We chose Muon as it appears to have a better scaling factor (Qiu et al., 2026).

	Parameter	Value
Model	Architecture	Qwen3
	Hidden Size	768
	Intermediate Size	3072
	Layers	12
	Vocabulary Size	20000
Training	Context Size	512
	Batch Size (in tokens)	32768
	Epochs	10
	Grdient Clipping	1
	Warmup Steps	1000
	Scheduler	Cosine
	Optimizers	Muon, AdamW
	Learning Rates	2e-2, 5e-3
	Weight Decays	0.1, 0.01
	Epsilons	1e-7, 1e-8
	Muon Momentum	0.95
	AdamW Betas	0.9, 0.95

Table 7: Hyperparameters used (except where otherwise indicated).

User: Translate the following text to `{tgt_lang}`. Don't output any other text:

`{text}`

Assistant:

User: Shorten the following text to `{sum_len}` words or less. Don't output any other text:

`{text}`

Assistant:

Figure 2: Prompts used for translation and summarization.

B Prompts

Figure 2 shows the exact prompts used for summarization and translation.