Pretraining Language Models with LoRA and Artificial Languages

Nalin Kumar and Mateusz Lango and Ondřej Dušek
Charles University, Faculty of Mathematics and Physics, Prague, Czechia
{nkumar,lango,odusek}@ufal.mff.cuni.cz

Abstract

Large language models (LLMs) require a substantial amount of training data, which contrasts with the data-efficient learning observed in humans. In our submission to the BabyLM Challenge, we address this disparity by proposing a parameter-efficient *pretraining* approach for language acquisition from limited data. Our approach involves initializing the model with token embeddings trained by a shallow model, followed by tuning the non-embedding parameters with non-linguistic data to introduce structural biases. Then, we freeze the resulting model and pretrain it on the 10M-token BabyLM corpus using LoRA adapters. Experiments on small corpora demonstrate that our approach improves upon classic pretraining of the entire model.

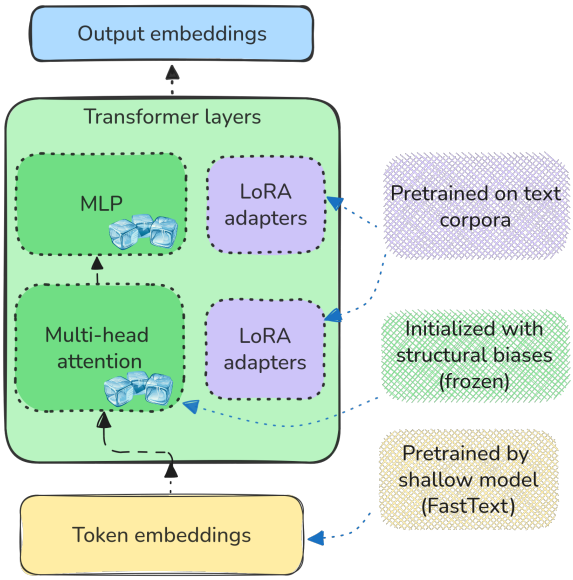

Figure 1: Overview of the proposed parameter-efficient pretraining.

1 Introduction

Large language models (LLMs) have shown impressive performance across a wide range of benchmarks, often rivaling human capabilities. However, their training requires far more data than humans need to acquire knowledge. To address the gap between the training efficiency of LLMs and that of a child, the BabyLM challenge provides an evaluation framework for developing data-efficient language models trained on human-scale training data of 10M–100M words (Warstadt et al., 2023; Hu et al., 2024; Charpentier et al., 2025).

This paper presents our submission to the strict-small track of the BabyLM Challenge, which aims to train high-performance language models using a corpus of just 10 million words. Our work focuses on developing parameter-efficient architectures for model pretraining, as smaller models typically achieve better results when trained on small datasets by reducing the risk of overfitting.

The proposed model is based on a BERT-like transformer architecture (Devlin et al., 2019), randomly initialized and pretrained on non-linguistic data (correct bracketing) designed to inject language-inspired structural biases into the model. After the initial pretraining stage, the entire model is frozen and further training on human-written texts is carried out using low-rank adapters (LoRA, Hu et al., 2021). This significantly reduces the number of parameters trained on text corpora compared to the original model. To further enhance parameter efficiency during pretraining, we also explore the use of shallow models for word embedding construction to initialize the embedding matrices.

An experimental evaluation on two popular benchmarks, BLiMP (Warstadt et al., 2020) and EWoK (Ivanova et al., 2024), reveals that models trained with the proposed parameter-efficient pretraining outperform those trained with standard pretraining of all parameters. Ablation experiments further demonstrate the benefits of non-linguistic

Proceedings of the First BabyLM Workshop, pages 525–530

data initialization and shallow models for embedding matrices. We publish our results and submitted models on HF repository.

2 Related Works

Word embeddings Early work demonstrating that models pretrained on free-text corpora can be useful for knowledge transfer across multiple deep learning tasks, was primarily focused on constructing word embeddings. Mikolov et al. (2013) proposed word2vec, which learns word embeddings using a skip-gram objective. Subsequently, the mathematical relation between word2vec and matrix decomposition was exploited to propose GloVe (Pennington et al., 2014). Later, the word2vec framework was extended to FastText (Bojanowski et al., 2017) that represents words as sums of character n-gram embeddings, with a hashing trick applied to improve parameter efficiency. All of these models are shallow and very fast to train, even on CPUs, yet they may provide valuable initialization points for embedding matrices in neural models (Kim, 2014). Note that embeddings constitute a significant fraction of the parameters of smaller models; for instance, in BERT (Devlin et al., 2019), roughly one-fifth of all parameters are devoted to token representations, so their good initialization may provide significant performance benefits.

Pretraining on artificial languages The effectiveness of pre-trained language models in performing downstream tasks has sparked considerable research interest in understanding the underlying reasons. This was often investigated using specially designed artificial languages. Papadimitriou and Jurafsky (2020) noticed that pretraining on non-linguistic data, such as MIDI music or sequences of pairs of matched integers, enhances the performance of language models on downstream tasks. Chiang and yi Lee (2022) further studied pretraining on integer strings, measuring its influence on the results on GLUE benchmark. They found that training on integer strings with the same unigram or bigram distributions as English words had a minimal effect on fine-tuning. Conversely, training on strings with stronger dependencies, e.g., containing groups of shuffled consecutive numbers or sequences of paired integers, resulted in significant improvements. These observations were also confirmed by Ri and Tsuruoka (2022). The injection of structural biases into models via pretraining on specific artificial languages was studied by Papadimitriou and Jurafsky (2023), whose experiments showed a reduction in perplexity of up to four times compared to random initialization.

Parameter-efficient fine-tuning The aim of parameter-efficient fine-tuning (PEFT) methods is to adapt large language models (LLMs) for downstream tasks by updating only a small subset of parameters, significantly reducing the computational and memory requirements. Techniques such as adapters (Chronopoulou et al., 2023), prefix tuning (Li and Liang, 2021), or fine-tuning only bias terms (Ben Zaken et al., 2022) have demonstrated competitive performance with full fine-tuning. A popular PEFT method is LoRA (Hu et al., 2021), which freezes the pretrained parameters of language models and approximates updates to weight matrices by a low-rank decomposition. To the best of our knowledge, such techniques were not previously used for language model *pretraining*.

3 Proposed Methodology

In this work, we propose a three-step approach for parameter-efficient pretraining on small text corpora: (1) using shallow model embeddings for better surface-level lexical representation, (2) initializing the language model with structural biases by pretraining it on artificial languages, and (3) pretraining the frozen model using LoRA adapters.

3.1 Initialization with pretrained embeddings

As token embedding matrices constitute a significant fraction of the parameters of small language models, we initialize them by optimizing the continuous skip-gram model objective (Mikolov et al., 2013) with a shallow linear model implemented in the FastText package (Bojanowski et al., 2017).

The word embedding model is trained using the same text corpus as the full language model, but with additional preprocessing. As FastText provides word embeddings and neural language models operate on tokens, the entire corpus is tokenised with a whitespace character introduced to split words into tokens. Next, standard FastText training is performed, with the word embedding size set to that of the neural model's input embeddings. The embedding layer of the language model is then initialised by the pretrained FastText embeddings.

3.2 Initialization with artificial languages

To initialize the model with structural biases, we experiment with pretraining on two artificial languages proposed by Papadimitriou and Jurafsky (2023): the nested parentheses language (NEST) and the crossing parentheses language (CROSS).

The NEST language has a vocabulary containing pairs of opening and closing tokens. Text is generated from left to right, with an opening token chosen with probability $p = 0.49$ and a closing token chosen with probability $p = 0.51$ in each iteration. If a closing token is selected, the most recently unmatched opening token is closed. An example sentence from the NEST language is:

$$1_(\; 24_(\; 24_) \; 67_(\; 39_(\; 39_) \; 67_) \; 1_)$$

The CROSS language operates on the same vocabulary as NEST; the difference is that the closing token can appear in any position after the opening token. Therefore, every NEST sequence is a correct CROSS sequence, but not vice versa, e.g.:

$$1_(\; 24_(\; 67_(\; 24_) \; 39_(\; 39_) \; 1_) \; 67_)$$

is a correct CROSS sentence. The text generation procedure of CROSS keeps the distribution of distances between the opening and closing tokens the same as in the NEST language.

The corpora generated in these two languages are applied for the initial pretraining of the transformer model with the standard masked language modeling objective. In this way, we can teach the model structural biases present in (non-)context-free grammars without using any language data and enable more efficient training on a small dataset.

3.3 Parameter-efficient pretraining

After initializing the transformer language model with pretrained embeddings and structural biases (on artificial languages), we freeze the weights of the entire model and inject LoRA's trainable rank decomposition matrices into each layer. The model is then trained with a standard masked language modelling (MLM) objective with default parameters from HuggingFace library (Wolf et al., 2020).

4 Experimental setup

4.1 Dataset

For pretraining on text data, we use the 10M-words version of BabyLM Corpus (Charpentier et al., 2025), comprising data sampled from 6 different domains. It includes OpenSubtitles (20%; dialogue from films), Simple English Wikipedia (15%; nonfiction), BNC (8%; dialogue), Project Gutenberg (26%; fiction & nonfiction), CHILDES (29%; dialogue), and Switchboard (1%; dialogue).

For all our experiments, we use the cased variant of the pretrained BERT tokenizer with a vocabulary size of 28k. The non-linguistic pretraining data consisted of 20,000 integer sequences following the grammar of artificial languages. Each sequence contained 512 tokens with a vocabulary size of 28k.

4.2 Autmomatic Evaluation Metrics

Models submitted to the BabyLM strict-small track are evaluated using a suite of automatic evaluation metrics: BLiMP (Warstadt et al., 2020), EWoK (Ivanova et al., 2024), GLUE (Wang et al., 2019), Entity Tracking (Kim and Schuster, 2023), WUG Adjective Nominalization (Hofmann et al., 2025), WUG Past Tense (Weissweiler et al., 2023), COMPS (Misra et al., 2023), Reading Cloze (de Varda et al., 2024), and AoA (Chang and Bergen, 2022). In this work, we report our results only for BLiMP and EWoK benchmarks.

4.3 Training details

FastText embeddings of dimension 768 were trained using the gensim library.[1] The training employed the skip-gram objective with a window size of 5 and was optimized for 5 epochs.

Pretraining on artificial languages was performed with the default HuggingFace Trainer hyperparameters, namely the AdamW optimizer with a learning rate of $5 \cdot 10^{-5}$ and a dynamic batch size. The optimization was performed for 25 epochs using the masked language modeling objective with a masking probability of 0.20.

We experimented with ranks $16, 64, 128, 256$ of LoRA (Hu et al., 2021), always setting the parameter α to twice the rank (i.e., $\alpha = 2 \cdot \text{rank}$). LoRA adapters were applied to all modules except the input and output embeddings and trained for 10 epochs. The pretraining setup otherwise followed the same hyperparameters described above.

4.4 Model Variants

All experiments use the BERT-base architecture (Devlin et al., 2019) as the underlying language model. In addition to testing the proposed approach, we perform experiments on various ablations to assess the contribution of each component.

[1] https://pypi.org/project/gensim/

Model			BLiMP	Supp.	EWoK	Avg
Embedding init.	AL init.	Pretraining				
BERT-base (Devlin et al., 2019) skyline			84.15	69.84	55.75	69.91
Model initializations						
Random	None	None	54.91	47.25	50.09	50.75
FastText	NEST	None	52.25	49.13	50.04	50.47
FastText	CROSS	None	57.51	50.05	**50.47**	52.67
Pretrained models						
Random	None	Standard	56.26	48.48	50.09	51.61
Random	None	LoRA	53.09	46.25	49.97	49.77
Random	CROSS	LoRA	52.66	45.32	50.11	49.36
Random	CROSS	LoRA + emb.	54.14	45.68	49.74	49.85
FastText	NEST	LoRA	55.99	51.73	50.02	52.58
FastText	CROSS	LoRA	**58.18**	**51.98**	50.38	**53.51**

Table 1: Evaluation results of trained language models on the 10M corpus with different initializations of embedding matrices (Embedding init.), initial pretraining on artificial languages (AL init.) and pretraining methods. LoRA + emb. indicates fine-tuning of LORA adapters together with input and output embedding matrices. LoRA is tested with the default rank of 16. Scores are measured on BLiMP, BLiMP Supplement (Supp.) and EWok benchmarks, with the Avg column showing an average of all three values.

LoRA rank	BLiMP	Supp.	EWoK	Avg
16	58.18	51.98	50.38	53.51
64	58.55	50.49	**50.43**	53.15
128	**60.96**	51.27	50.25	54.16
256	60.20	**53.21**	50.10	**54.50**

Table 2: Results of our approach (initialized by FastText and CROSS language) with different ranks of LoRA matrices (see Table 1 for scores).

Model Initializations We evaluate model performance without pretraining on linguistic data. Specifically, we evaluate the performance of completely randomly initialized language model, as well as the models initialized by FastText and pretrained on CROSS and NEST artificial languages.

Pretrained Models We also investigate several variants of pretrained models. The primary baseline is a transformer model trained in the standard way: weights are randomly initialized, and pretraining updates all model parameters. We then test models with LoRA adapters and word embeddings initialized either randomly or with FastText. Similarly, variants initialized with the NEST and CROSS artificial languages are evaluated.

5 Results

Table 1 presents the automatic evaluation scores on BLiMP, BLiMP Supplement (Supp.), and EWoK. Among the compared settings, FastText-CROSS-LoRA achieves the best performance, showing a gain of approximately four points over its counterpart initialized with random embeddings (Random-CROSS-LoRA). Overall, initializing the model with FastText embeddings consistently outperforms random initialization. Artificial language pretraining appears beneficial only in the CROSS setting, while configurations using NEST tend to degrade performance. Interestingly, the model pretrained only on artificial languages with FastText initialization obtained better performance than the standard pretraining on text data. LoRA-based pretraining yields slightly better results on BLiMP and BLiMP Supplement benchmarks.

Since LoRA introduces only a small number of trainable parameters and the default rank of 16 is designed for fine-tuning only, the model may not have sufficient capacity for pre-training and thus underfit on the BabyLM corpus. To address this, we experimented with higher LoRA matrix ranks (see results in Table 2). For BLiMP, performance generally improves as the rank increases, but slightly drops at a higher value, 256. In the case of BLiMP Supp., the highest LoRA rank yields the best results. By contrast, similar to BLiMP Supp., performance on EWoK does not show any consistent correlation with increasing rank.

6 Summary

This paper presents a parameter-efficient approach for pretraining language models on small text corpora. The main innovations include the usage of artificial languages to induce structural biases, using shallow models for matrix embedding initialization and pretraining a large model with LoRA adapters.

Limitations

This paper was limited in testing different configurations of trained models and it is highly probable that the training parameters used were not optimal.

Acknowledgments

This work was supported by the European Research Council (Grant agreement No. 101039303, NG-NLG) and Grant Agency of Charles University (Grant No. 302425), and used resources of the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2018101).

References

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. Babylm turns 3: Call for papers for the 2025 babylm workshop. *Preprint*, arXiv:2502.10645.

Cheng-Han Chiang and Hung yi Lee. 2022. On the transferability of pre-trained language models: A study from artificial datasets. *Preprint*, arXiv:2109.03537.

Alexandra Chronopoulou, Matthew Peters, Alexander Fraser, and Jesse Dodge. 2023. AdapterSoup: Weight averaging to improve generalization of pre-trained language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2054–2063, Dubrovnik, Croatia. Association for Computational Linguistics.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 english sentences, aligned with existing eeg and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

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.

Valentin Hofmann, Leonie Weissweiler, David R Mortensen, Hinrich Schütze, and Janet B Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Anna A Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H Clark, Carina Kauf, Jennifer Hu, RT Pramod, Gabriel Grand, and 1 others. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Preprint*, arXiv:1408.5882.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *Preprint*, arXiv:2101.00190.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Isabel Papadimitriou and Dan Jurafsky. 2020. Learning Music Helps You Read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6829–6839, Online. Association for Computational Linguistics.

Isabel Papadimitriou and Dan Jurafsky. 2023. Injecting structural hints: Using language models to study inductive biases in language learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8402–8413, Singapore. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ryokan Ri and Yoshimasa Tsuruoka. 2022. Pretraining with artificial language: Studying transferable knowledge in language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7302–7315, Dublin, Ireland. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. 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.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.