BabyLM Challenge: Experimenting with Self-Distillation and Reverse-Distillation for Language Model Pre-Training on Constrained Datasets

Aakarsh Nair, Alina Hancharova, Ali Gharaee, Mayank Kumar
Seminar für Sprachwissenschaft, Eberhard Karls Universität Tübingen
Keplerstraße 2, 72074 Tübingen, Germany
{first.last}@student.uni-tuebingen.de

Abstract

Language models (LMs) exhibit significant data inefficiency compared to human learners. A child is able to master language while consuming less than 100 million words of input, while language models require orders of magnitude more tokens during training.

Our submission to the BabyLM Challenge utilizes a combination of self-distillation and reverse-distillation to train a sequence of ensemble models with improved training characteristics on a fixed-size 10 million-word dataset.

Self-distillation is used to generate an ensemble of models of a certain fixed size, while reverse distillation is used to train a more expressive larger model from a previously trained generation of relatively smaller models, while largely preserving learned accuracy.

We find that ensembles consisting of two smaller models and one identical born-again model serves as an ideal ensemble for each trained generation of model size. We demonstrate that, although our method is not novel, it provides consistent and modest performance improvements on the BLiMP and GLUE benchmarks.

1 Introduction

Brown et al. (2020) have demonstrated that large language models (LLMs) have impressive capabilities in various natural language processing tasks.

Moreover, the availability of open-source models such as Llama-2 (Touvron et al., 2023) has enabled researchers to fine-tune pre-trained models for application-specific tasks.

Pre-training language models, however, remain out of reach for most researchers due to prohibitive computing and data requirements. For example, state-of-the-art models like Chinchilla (Hoffmann et al., 2022) and GPT-2 (Radford et al., 2019) are trained on approximately 1.4 trillion words and 200 billion words, respectively. This is in sharp contrast with the 100 million words which a human teenager might see during their lifetime (Warstadt and Bowman, 2022).

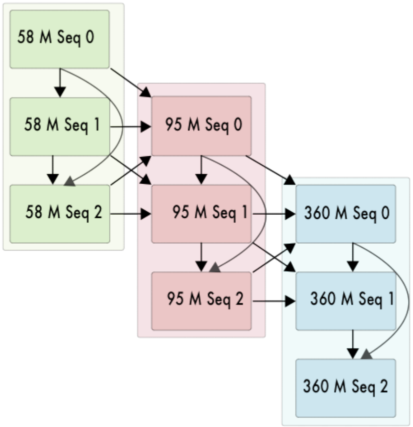


Figure 1: We train an expanding series of models using a moving window ensemble containing the previously trained models (left to right) as teachers. The model with sequence number 1 is trained on two predecessor models of smaller size and one of the same size. While models with sequence numbers 0 and 2 are trained in a uniform ensemble of smaller-sized and equal-sized models respectively

The BabyLM Challenge is a shared task for CoNLL 2024 (Choshen et al., 2024), meant to incentivize research into optimization of training on constrained datasets. In the *strict-small* track of this challenge, researchers are limited to using a 10 million word text-only dataset to be used for pre-training.

In this paper, we explore the performance of *decoder-only* architectures using *self-distillation* and *reverse-distillation* starting from a base model trained on the same dataset. Following the training protocol described in Figure 1.

For our base model, we chose to start with the preceding year's decoder-only model *BabyLlama*

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 28–36
November 15-16, 2024 ©2024 Association for Computational Linguistics

(Timiryasov and Tastet, 2023) and retrained on it on this year's challenge dataset.

We subsequently trained an ensemble of teachers of increasing sizes using *self-distillation* (SD) and *reverse-distillation* (RD), attempting to characterize the effect of model size and ensemble structure on the model's performance while keeping the dataset constant.

During Knowledge Distillation, a teacher network, usually a higher capacity network is used to train a student network, which may be of lower capacity (Hinton et al., 2015). The emphasis of Knowledge Distillation has typically been on model compression, where a student network is expected to be a more compact representation of its teachers.

In self-distillation, as described by Furlanello et al. (2018) in their work on Born-Again Neural Networks, one observes that a neural network of a given size can be re-initialized and trained with guidance from previously trained instances of itself. This process results in a student network that can maintain or even improve upon the performance of its teacher networks. Reverse distillation expands on this idea by training a student network that is larger than its teacher network, potentially enabling better generalization and the capacity for further training.

2 Related Work

Knowledge distillation (Hinton et al., 2015), a technique central to our work, has emerged as a popular approach for transferring knowledge from large models to smaller, more efficient ones. Furlanello et al. (2018) introduced the concept of "Born Again Neural Networks," where neural networks are trained using the predictions of an already-trained model, illustrating the potential of self-distillation. Gou et al. (2021) provided a comprehensive survey of various knowledge distillation techniques, categorizing them based on model types and applications and demonstrating their use in optimizing neural networks for various tasks, including language modeling.

We build on work by Timiryasov and Tastet (2023), which contributed to the area by exploring knowledge distillation from an ensemble of teacher models trained on small datasets, achieving competitive results without performance degradation. Whereas *BabyLlama* compressed large models into a smaller model, we attempt to use born-again ensembles of these smaller models to learn successively larger models. We find our techniques largely preserve and improve the base model's accuracy. While *BabyLlama* compressed model outperforms its teachers, our model expansion preserves these gains and allows us to continue learning with larger models. The larger expanded models have also been found to be more amenable to fine-tuning downstream tasks.

3 Methodology

3.1 Models

Feature	58M	95M	360M
Hidden Layers	16	10	24
Attention Heads	8	12	8
Hidden Size	512	768	1024
Intermediate Size	1024	2048	3072
Teacher Quantization	-	-	int8

Table 1: Model Variants and Architecture Details

We trained a series of decoder models with increasing sizes—58M, 95M, and 360M—following the training protocol outlined in Figure 1. Each model size includes a sequence of three models, all based on the decoder-only Llama architecture (Vaswani, 2017). The architectural details for each model variant are summarized in Table 1.

Sequence zero for a given model size is trained using a teacher ensemble, which consists of three models strictly smaller than the current model. Sequence one is trained with two smaller models and one model of the same size. Sequence two is trained with two models of the same size and one smaller model. For each model size from 95M onward, three teacher models are used. However, the initial 58M model is trained in a strictly born-again sequence.

Our base model is 58 M Sequence 0 is the base model (Timiryasov and Tastet, 2023), which we trained using this year's dataset from scratch. (Note that this starting model performs below the later released contest baseline *BabyLlama* model). We apply identical prepossessing and tokenization as *BabyLlama* Model on the 10 million word dataset, provided by BabyLM challenge organizers.

3.2 Hardware

The models 58M and and 95M were trained on Nvidia T4 GPU, while the 360 M models were trained on Nvidia A100 where the 360 M teachers were quantized down to `int8` when used for inference during their teaching phase.

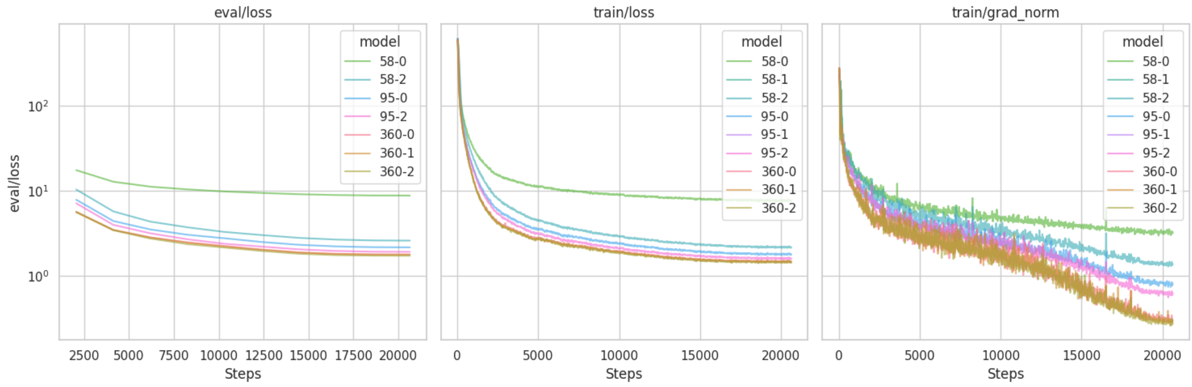


Figure 2: Evaluation and training loss along with gradient norms for models in the sequence. We note that models later in the teaching sequence and larger models have steeper decline losses than models earlier in the sequence.

3.3 Loss Function

We use the distillation trainer to construct teacher ensembles, with a weighted sum of original cross-entropy loss for training labels and a distillation loss for matching the teacher ensemble's targets from Timiryasov and Tastet (2023).

$$L = \alpha L_{\text{cross-entropy}} + (1 - \alpha) L_{\text{Kullback Leibler}} \quad (1)$$

We vary the composition of an ensemble of teachers as described previously. Distillation trainer parameters were chosen as in BabyLLama, with a sequence length of 128, a temperature of 2.0, and $\alpha = 0.5$. Trainer hyper-parameters are listed in Table 7.

4 Results

We evaluated the models on three benchmarks: GLUE (Wang, 2018), BLiMP (Warstadt et al., 2020), and EWoK (Ivanova et al., 2024). For the GLUE benchmark, an additional fine-tuning phase was included to enhance the model's task-specific performance. Detailed results are provided in Appendix A.

4.1 Training

Figure 2 illustrates the training dynamics observed for each model in the sequence. Successive models and those of larger sizes consistently displayed lower validation losses compared to their predecessors. Training losses and gradient norms also decreased more sharply in later sequence models. While validation loss did not always correlate with improved performance across all benchmarks, models later in the sequence generally performed better on several tested benchmarks.

4.2 BliMP

The results of the BliMP benchmark for our student/teacher models can be seen in Table 2. We note that sequences of larger models tend to perform better on average on BLiMP tasks than the smaller models. We note that Sequence 1 tends to perform better than Sequence 0 for model sizes 95 and 360. We hypothesize that this effect might be due to smaller models, as teachers might have regularizing effects on teaching labels, while the Sequence-0 model of the same size might help in training the Sequence-1 model during training. Further ablation studies would required to confirm the optimal ensemble combination of teachers for a model.

We note that the lower validation loss in successive generations does not capture the drop in BLiMP accuracy which we note between Sequence 1 and Sequence 2 of model size. Thus cross-entropy and divergence loss are failing to capture nuances being tested in the benchmarks.

Table 5 shows the results on the 14 BLiMP subtasks. In Figure 5) We plotted the accuracy of the BLiMP sub-tasks, which had the highest variance in model accuracy. We note that larger models are improving in accuracy; however, for anyone subtask, the improvements are not strictly monotonic. For example, the wh_island subtask performance has two peaks in accuracy: one for model 95 M model of Sequence 1 and another for 360 M of Sequence 2.

4.3 GLUE

Table 3 provides a detailed breakdown of the model performance on each of the various GLUE subtasks. GLUE benchmarks involve an initial task fine-tuning phase before the benchmark metrics are

Model Size	Sequence #	BLiMP	Sup.
58 M	0	0.68709	0.5637
58 M	1	**0.69058**	0.56742
58 M	2	0.69051	**0.58007**
95 M	0	0.68926	0.57322
95 M	1	**0.69395**	**0.57396**
95 M	2	0.69147	0.56693
360 M	0	0.69605	**0.58694**
360 M	1	0.69815	0.58042
360 M	2	**0.70102**	0.58267

Table 2: Model accuracy by size and iteration number on the blimp evaluation. We note that accuracy improves with model size and that iterations that have two smaller prior models in the teacher ensemble have higher accuracy for a given model size. Supplementary runs are also provided for reference; however, we only observe a trend of larger models being better in these results.

computed. The details of the list of fine-tuning parameters for GLUE that are used are provided in Table 6. Notably, due to computational constraints, the models were fine-tuned for three epochs prior to evaluation.

Figure 3 shows the qualitative performance of all nine of our trained models. We observe that 6 of the 11 tasks in GLUE models performed at approximately the same level. However, models 360-1 and 360-2 show significant improvement in fine-tuned accuracy on tasks in `wsc`, improving from 37% baseline performance to 48% and 50% respectively. While models 95-1 and 95-2 roughly double the baseline accuracy to approximately 60%. As in BLiMP, we observe that task performance is not monotonically increasing.

Other modest improvements are seen for models 95-1 and 95-2 task `rte`: from 50% in baseline accuracy to 53% for both of them. The best-performing model on `rte` **360-0** has both these models in its parent model and can preserve and improve upon their accuracy.

Model **360-0** is the best performing model on tasks `cola`, `multirc`, `rte`. While models $95-1$, $95-2$, $360-1$, $360-2$ have higher average performance. Notably, the majority of the models outperform the chosen baseline model in average performance.

In both model classes 95 and 360, the sequence 1 models have the highest average performance. Thus, we hypothesize, as in the case of BliMP, that having two smaller models along with the same sized model in the ensemble allows sequence 1 models with more excellent stability, with smaller models having a regularizing effect on learned labels, thus allowing sequence 1 models to preserve knowledge of previous sequences. Thus, further investigation into a measurement of *catastrophic forgetting* between model sequences is required (Kemker et al., 2018).

4.4 EWoK

Finetuning on the EWoK benchmark doesn't show any significant progress among models. The average accuracies for models have differences only at hundredths of a percent (See Table 4 and Figure 4). Further analysis of this benchmark is not included in our results.

5 Conclusion

In this study, we have shown that we can train an ensemble of born-again teacher networks and use the ensemble of teachers to train larger student models. We find that having a model of the same size while having two models of smaller sizes in the ensemble leads to consistent improvements in the BLiMP benchmark. Similar improvements are also noted on GLUE benchmarks, which included an intermediate finetuning step.

We note that the accuracy of a smaller model is not lost in the reverse distillation process, thus allowing us to continue training with a larger models.

For several of the benchmark tasks, however, we observe that improvements are non-monotonic but trend upward. Thus, knowledge-distillation for student models is not consistently noise-free.

This self-distillation and reverse-distillation process can be repeated to grow the size of our ensembles. With larger models more amenable to finetuning.

Further work is needed to quantify the limits of this method of improvement compared to directly training a large network and distilling it down to a smaller model. Moreover, further work is required to quantify measures of catastrophic forgetting, as validation loss is often not predictive of benchmark performance and particular sub-task/skill.

6 Limitations

This study used the BablyLM dataset out of the box, but it could have benefited from more straightforward datasets available in a more consistent format. Further pre-processing and curriculum design would possibly provide improvements over currently applied methods.

Although the inspiration for this paper was based on a hypothesis about a sequence of teaching selves from (Minsky, 1988). The methods employed in this paper are not guided by strong priors of biological plausibility.

In contrast to human learning which often involves multiple modalities including real-world interactions, visual and audio perception in the formation of the language faculty such grounding was not utilized by our current method. Thus, no understanding of phonetics, visual concepts, or intuitive physics was needed to bootstrap our model.

The sequence of teachers employed in this paper trades off lack of data availability with the computing required to train each subsequent round of teachers from the ground up; further study is required to investigate if prior knowledge of teachers can be incorporated in a less compute-intensive manner, such that skills learned by teachers are not lost in subsequent rounds of self-distillation and reverse distillations.

While most metrics were preserved in such subsequent rounds, some metrics did suffer from distillation and only recovered further down in the sequence.

Moreover, the further down the sequence one proceeds with increasing the model size, one runs into computational challenges. Thus, we were required to use quantization to accommodate larger models on our compute node. We also limited the number of training and fine-tuning epochs to stay within resource constraints.

Further study is also required to understand the effects of chosen hyper-parameters as we increase the size of the teachers in later stages of inference.

Finally, this approach depends on the availability of a distilled smaller model as a starting point for training. Further investigation is required on how distillation back down to smaller models from our larger models will preserve the newly learned skills and if auto-regressive training of our sequences is thus possible.

A Appendix

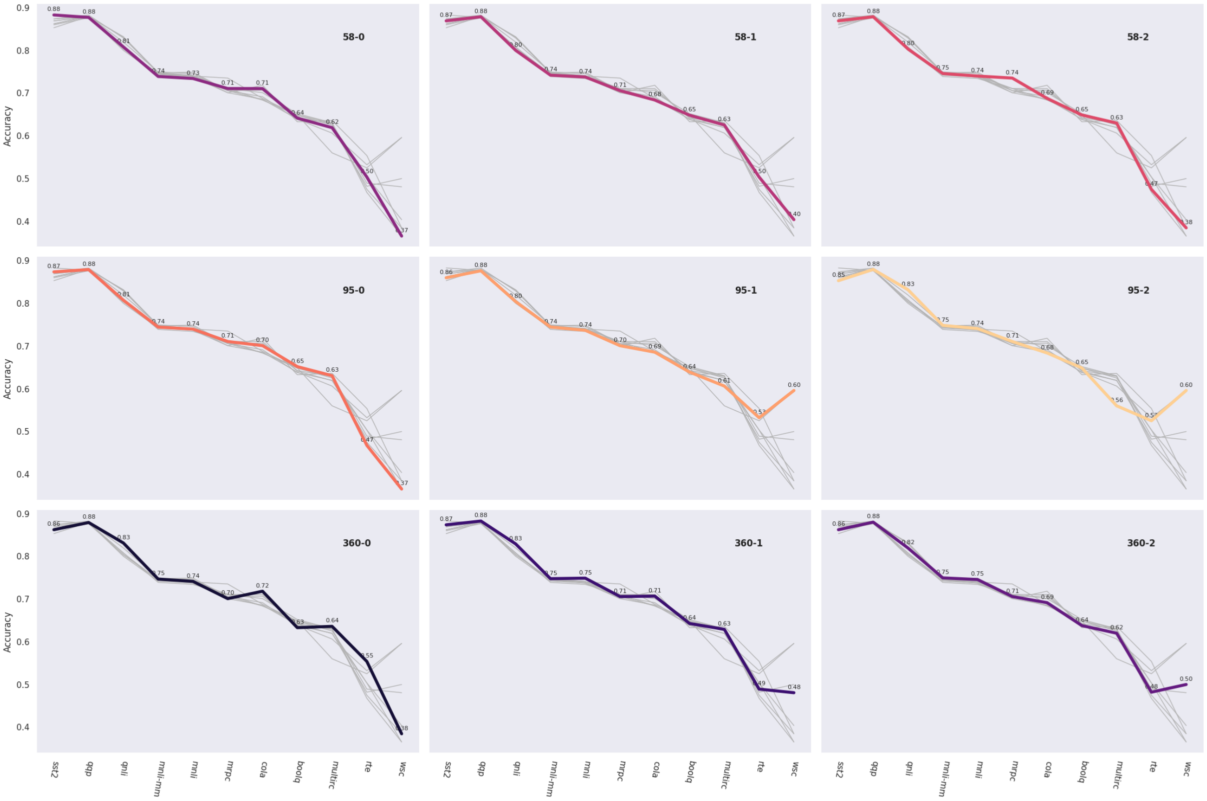
Figure 3 shows qualitative results on GLUE benchmarks. See Table 3 for quantitative results on GLUE. The finetuning parameters used for GLUE are listed in Table 6.

Similar qualitative and quantitative results on EWoK can be seen in Figure 4 and Table 4.

For BLiMP, we visualize subtasks with the highest variance across models in Figure 5 while Table 5 provides a full quantitive breakdown by subtasks.

Lastly Table 7 lists the trainer hyper-parameters used to construct the ensembles.

Figure 3: GLUE results for 9 models. All models were fine-tuned with standard params given by BabyLLM organizers except the number of epochs parameter, which was set to 3



model	qqp	sst2	qnli	mnli-mm	mnli	mrpc	cola	boolq	multirc	rte	wsc	avg
58-0*	0.8773	**0.8830**	0.8082	0.7390	0.7343	0.7108	0.7107	0.6416	0.6192	0.5036	0.3654	0.6903
58-1	0.8788	0.8693	0.8001	0.7421	0.7378	0.7059	0.6839	0.6483	0.6254	0.5036	0.4038	0.6908
58-2	0.8789	0.8693	0.8034	0.7459	0.7400	**0.7353**	0.6877	**0.6489**	0.6299	0.4748	0.3846	0.6908
95-0	0.8791	0.8739	0.8075	0.7445	0.7398	0.7108	0.7011	0.6520	0.6308	0.4676	0.3654	0.6884
95-1	0.8764	0.8601	0.8042	0.7447	0.7370	0.7010	0.6858	0.6391	0.6064	0.5324	**0.5962**	**0.7076**
95-2	0.8795	0.8532	**0.8320**	0.7486	0.7410	0.7108	0.6839	**0.6489**	0.5602	0.5252	**0.5962**	0.7072
360-0	0.8792	0.8624	0.8313	0.7467	0.7414	0.7010	**0.7184**	0.6330	**0.6361**	**0.5540**	0.3846	0.6989
360-1	**0.8827**	0.8739	0.8291	0.7478	**0.7490**	0.7059	0.7069	0.6428	0.6291	0.4892	0.4808	0.7034
360-2	0.8801	0.8624	0.8195	**0.7496**	0.7457	0.7059	0.6916	0.6379	0.6200	0.4820	0.5000	0.6995

Table 3: Performance of models on GLUE tasks, sorted by mean accuracy. The models were finetuned for 3 epochs for each of the Glue Benchmarks. 58-0 is considered the baseline model with which we compare.

Model	Ewok Average Accuracy
58-0	**0.5041**
58-1	0.5018
58-2	0.5002
95-0	0.4959
95-1	0.5001
95-2	0.5021
360-0	0.5008
360-1	0.5017
360-2	0.5013

Table 4: No significant improvement was found on EWOK tasks. Overall accuracy stayed the same, with minor variations downwards.

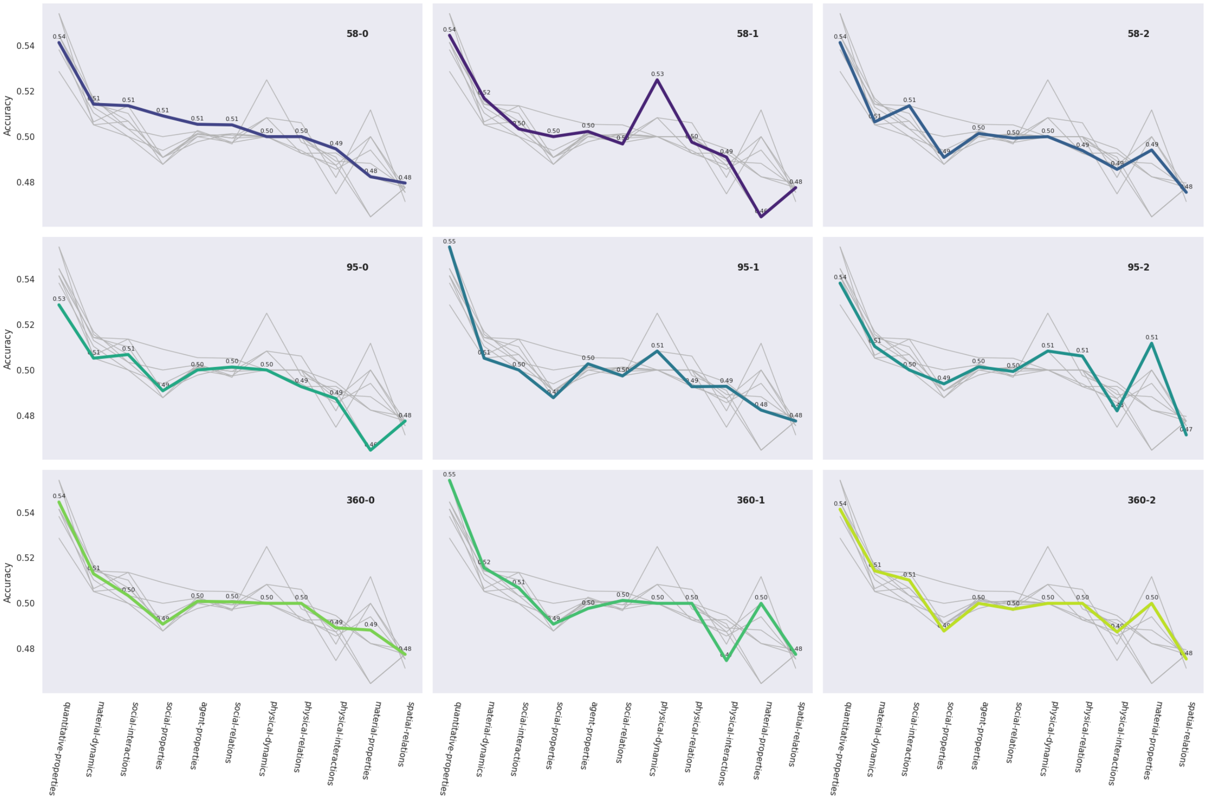


Figure 4: Ewok results for 9 models. Standard parameters were used to run Ewok evaluations.

Subtask	58-0	58-1	58-2	95-0	95-1	95-2	360-0	360-1	360-2
coordinate_structure_constraint_complex_left_branch	**0.292**	0.266	0.234	0.245	0.228	0.235	0.233	0.233	0.245
existential_there_quantifiers_2	0.427	0.403	0.337	0.367	0.361	0.341	0.387	**0.457**	0.437
irregular_past_participle_adjectives	0.976	0.917	0.896	0.965	0.953	0.947	0.968	0.974	**0.979**
left_branch_island_echo_question	0.559	0.614	0.546	**0.581**	0.420	0.427	0.528	0.445	0.553
left_branch_island_simple_question	**0.479**	0.456	0.427	0.417	0.420	0.423	0.467	0.438	0.447
matrix_question_npi_licensor_present	0.099	0.131	0.115	0.105	**0.239**	0.230	0.104	0.144	0.141
npi_present_1	0.230	0.268	0.274	0.275	0.265	0.276	0.312	0.283	**0.315**
npi_present_2	0.235	0.310	0.344	0.362	0.317	0.328	0.362	0.365	**0.376**
only_npi_licensor_present	0.821	0.997	0.997	**1.000**	0.994	0.986	0.985	0.965	0.992
only_npi_scope	0.508	0.547	0.503	0.485	0.591	**0.601**	0.544	0.517	0.519
principle_A_c_command	0.505	0.558	0.532	0.529	0.558	0.523	0.554	0.556	**0.570**
principle_A_domain_2	**0.742**	0.678	0.714	0.730	0.675	0.711	0.702	0.692	0.705
superlative_quantifiers_1	0.851	0.764	0.838	0.831	**0.888**	0.839	0.857	0.815	0.849
superlative_quantifiers_2	0.610	0.644	0.680	0.612	0.773	0.795	0.688	**0.831**	0.768
wh_island	0.526	0.506	0.523	0.546	**0.601**	0.533	0.600	0.598	**0.601**

Table 5: Break down of BliMP accuracy by subtasks. Results on BLiMP filtered subtasks for different models. We note that later models tend to perform better. With a handful of metrics losing performance.

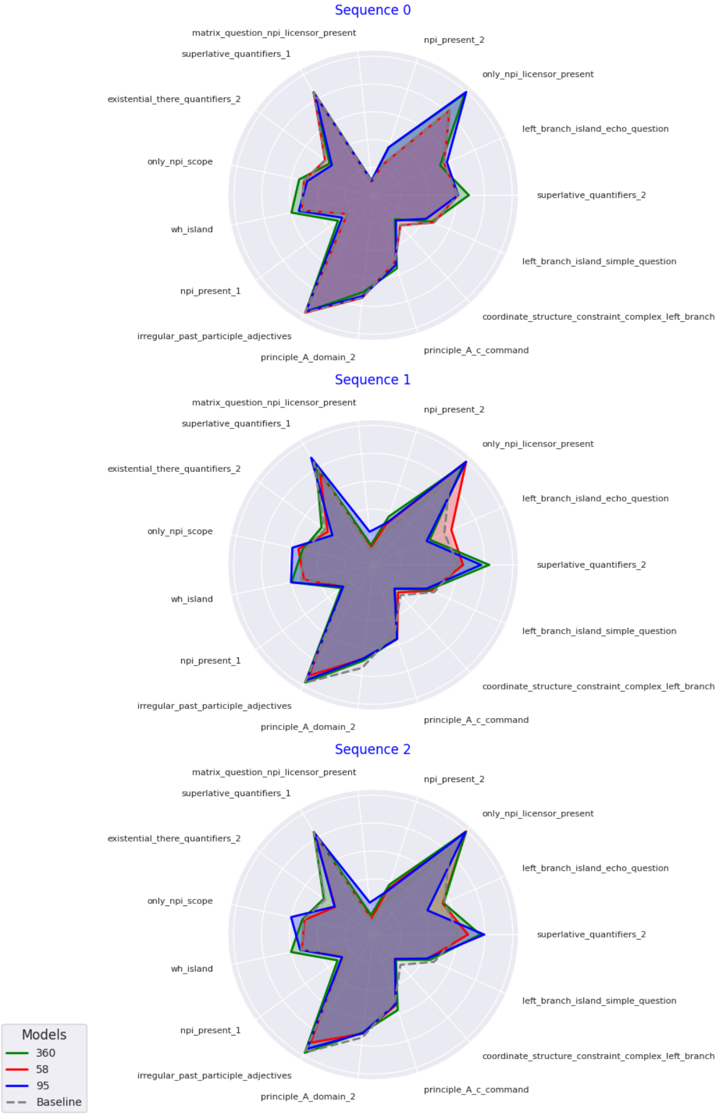


Figure 5: Blimp results for 9 models, grouped by sequence. All models were fine-tuned with standard parameters given by BabyLLM organizers except the number of epochs parameter, which was set to 3. We show the sub-tasks which have the highest variance across the models.

Fine Tuning Hyper-parameters	Value
Learning Rate	5e-5
Patience	3
Batch Size	64
Max Epochs	**3**
Seed	12

Table 6: GLUE fine-tuning hyper-parameters, due to computational cost limitations, fine-tuning was only performed for 3 epochs.

Trainer Hyperparameters	Value
Seed	42
Learning Rate	0.00025
Train Batch Size	64
Eval Batch Size	8
Optimizer	Adam β=(0.9, 0.999), ϵ=1e-08
LR Scheduler Type	cosine
LR Scheduler Warmup Steps	200
Number of Epochs	10

Table 7: Trainer Hyperparameters

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Leshem Choshen, Ryan Cotterell, Michael Y Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [call for papers] the 2nd babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2404.06214*.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. In *International conference on machine learning*, pages 1607–1616. PMLR.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

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

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyurek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2024. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. 2018. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Marvin Minsky. 1988. *Society of mind*. Simon and Schuster.

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

Inar Timiryasov and Jean-Loup Tastet. 2023. Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. *arXiv preprint arXiv:2308.02019*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Alex Wang. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Warstadt and Samuel R Bowman. 2022. What artificial neural networks can tell us about human language acquisition. In *Algebraic structures in natural language*, pages 17–60. CRC Press.

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.