DATA EFFICIENT PRE-TRAINING FOR LANGUAGE MODELS: AN EMPIRICAL STUDY OF COMPUTE EFFICIENCY AND LINGUISTIC COMPETENCE

Andreas Paraskeva*
Leiden University
Leiden, Netherlands

Max Johannes van Duijn
Leiden University
Leiden, Netherlands

Maarten de Rijke
University of Amsterdam
Amsterdam, Netherlands

Suzan Verberne
Leiden University
Leiden, Netherlands

Jan N. van Rijn
Leiden University
Leiden, Netherlands

ABSTRACT

Training large language models is compute- and data-intensive, limiting optimisation and low-resource training, and increasing environmental impact. This paper examines pre-training effectiveness of language models of different sizes on two small, curated datasets and evaluates (i) linguistic competence and (ii) compute efficiency. The datasets are TinyStories, a collection of ChatGPT-generated children's stories, and BabyLM, a small, open-domain dataset. We perform experiments with increasing amounts of data (yielding a learning curve) and size-variants of a Llama-based, decoder-only architecture. We evaluate the pre-trained models on downstream tasks from the BLiMP and GLUE benchmark suites. We find that models trained on BabyLM outperform those trained on TinyStories on formal linguistic competence, but not on functional linguistic tasks. Models pre-trained on BabyLM yield more consistent performance results, as indicated by lower variance across random seeds. We also find that small data samples are representative of the model's ultimate performance, which can aid the early selection of promising candidate models. These findings emphasise the potential of pre-training on small, curated datasets for data-efficient pre-training in resource-constrained settings. Further work that includes additional datasets and model architectures is needed to extend the scope of these findings.

1 INTRODUCTION

Large language models (LLMs) based on the transformer architecture (Vaswani et al., 2017) have made remarkable progress in achieving linguistic competence over the past years. Linguistic competence is characterised as twofold, involving both the mastery of grammatical and structural rules (*formal competence*) and the ability to use and reason with language in real-world contexts (*functional competence*) (Mahowald et al., 2023; Eldan & Li, 2023).

Following evidence that linguistic competence and other capabilities scale with model size (Wei et al., 2022), we have recently seen a tendency to build ever larger LLMs in terms of numbers of parameters and training data size. This has led to increasing concerns about the data hunger of current models for reasons of scaling limits (Sutskever, 2024), but also environmental footprint (Dhar, 2020) and the impossibility of checking immense datasets for quality, bias, and copyright violations (Verberne, 2024). In light of these concerns, researchers have explored various ways to build smaller models with no or little loss of linguistic competence. The majority of work in this area focuses on model architecture (Gu & Dao, 2023) and posthoc interventions such as fine-tuning (Li & Liang, 2021), model quantisation (Xiao et al., 2023), and constrained decoding (Beurer-Kellner et al., 2024). Yet, there is also increasing attention to aspects

*a.paraskeva@liacs.leidenuniv.nl

that influence the pre-training procedures (Gururangan et al., 2020) and the quality and nature of the training data.

In this work, we investigate the use of **small curated datasets** in the pre-training phase of small-scale decoder-based transformers. We compare models pre-trained on two intristically different datasets: (i) the **TinyStories dataset**, containing ChatGPT-generated narratives with a simple vocabulary, typically understandable for children around the age of 4 (Eldan & Li, 2023), versus (ii) the **BabyLM dataset**, encompassing diverse human-produced data from publicly available domains, such as child-directed speech, dialogues, Wikipedia articles, and books (Choshen et al., 2024).

We experiment with Llama-based decoder-only transformer models, varying their total trainable parameters by adjusting key architectural components via the Hugging Face interface (Wolf et al., 2020). The models being evaluated are categorised by their total number of parameters: 17 million, 58 million and 91 million. We propose an experimental method to analyse the performance of the models across increasing dataset sizes, generating learning curves (Mohr & van Rijn, 2024; Viering & Loog, 2023) for our models. We report on how the performance of the models develops when using more tokens.

Our experimental methodology covers two stages of language model development: pre-training and supervised fine-tuning to a specific task. Given the number of models, random seeds, datasets, and anchors, we pre-train a total of 180 model instances. Each instance then undergoes fine-tuning for individual tasks. This approach demands substantial compute and storage, as model weights are stored separately per task.

To achieve a structured approach for comparing models trained on different data sources, we measure formal linguistic competence with the BLiMP benchmark suite (Warstadt et al., 2020) and functional linguistic competence on a set of GLUE downstream tasks (Wang et al., 2019b). Additionally, analysis of token-based measures allows us to evaluate compute efficiency for each of the two datasets.

Our work makes three key contributions.

1. We propose an **experimental pipeline**[1] for evaluating language model training on token-based anchors across datasets. An anchor is a fixed point of reference during the training process, where the performance of the model under investigation can be measured and used for comparison. In our specific case, we use anchors of the full dataset, thereby sampling incrementally bigger portions of the data in terms of number of tokens. We pre-train models (from scratch) at each *token-based anchor* and track performance through evaluation, forming a learning curve. This pipeline assesses both formal and functional linguistic competence.
2. We show that models trained on **BabyLM** achieve significantly higher performance on formal linguistic competence and show lower variance and more consistent performance increase across anchors than models trained on **TinyStories**. However, the performance gap is smaller for functional linguistic competence. Since the difference in formal competence can be attributed to the fact that linguistic structures and vocabulary are 'simple-by-design' in **TinyStories**, these findings highlight the potential of narrative datasets in pre-training language models and suggest their applicability in curriculum learning.
3. We demonstrate that early performance metrics can be used in model selection, allowing us to discard unpromising candidate models.

These contributions amount to a better understanding of how dataset size, complexity, and sampled domains influence linguistic competence when pre-training language models, and how they impact practical applications of compute efficiency and model deployment.

2 Related Work

Recently, significant research attention has been given to data-centric problems in the training of LLMs. Various datasets and approaches have been developed that take inspiration from child development (Huebner et al., 2021; Eldan & Li, 2023; Feng et al., 2024; Choshen et al., 2024). Humans

[1] GitHub page: `https://github.com/ADA-research/data-efficiency`.

become fluent speakers after interacting with an amount of language 'data' that is several orders of magnitude smaller than even mid-sized LLMs (Warstadt & Bowman, 2022). This led to work on pre-training of LLMs to explore various characteristics of children's learning environments, including the use of child-directed speech, children's stories, and multi-modal input (Warstadt et al., 2023).

Eldan & Li (2023) introduce a synthetic, ChatGPT-generated dataset called **TinyStories**, consisting of short stories with a simple vocabulary, typically comprehensible by 3 to 4 year-old children. The authors showcase the ability of small language models, once trained on this dataset, to generate coherent and typically grammatically correct stories. However, their evaluation focuses exclusively on the assessment of generated text (stories) by these models and their respective quality of grammar, coherency, creativity, and other qualitative measures of generated text.

The aim of the **BabyLM** challenge (Choshen et al., 2024) is to optimise the pre-training stage of language models with small data.[2] The challenge has released a small dataset for researchers to train their models on, which is much smaller than commonly used datasets for language model training. The size is specified by word count (there is a 10-million and a 100-million version), and the datasets comprise components from various domains representing a diverse vocabulary. The challenge's evaluation incorporates state-of-the-art benchmarks, such as GLUE and BLiMP, to assess linguistic competence (see Section 3.3).

Although both **TinyStories** and **BabyLM** are small datasets for language model pre-training, they differ in two important respects: (i) **TinyStories** exclusively contains stories while **BabyLM** is compiled from a variety of open-domain genres, and (ii) **TinyStories** consists of LLM-generated text while **BabyLM** consists of traditional, human-written data. Our work compares small language models pre-trained on the **TinyStories** and **BabyLM** datasets, focusing on two aspects: linguistic competence and compute efficiency. By performing qualitative and quantitative evaluations, we aim to provide a more thorough understanding of these curated datasets' abilities for language model training while considering the underlying computational costs.

In prior work, Feng et al. (2024) investigate whether child-directed speech is beneficial for training language models. They have generated a synthetic dialogue data, named TinyDialogues, and compared its effectiveness against natural child-directed speech (CHILDES (MacWhinney, 2000)) and other domains of dataset like OpenSubtitles (Lison & Tiedemann, 2016), Wikipedia (Xu & Lapata, 2019) and BabyLM (Choshen et al., 2024). Their evaluation focuses on semantic and syntactic knowledge (covering functional linguistic competence), with a partial focus on the effectiveness of global and local ordering. The study shows that synthetic child-directed data outperformed natural child-directed data. While curriculum learning did not significantly improve the final performance of language models, maintaining the logical order of dialogues (i.e., retaining the local ordering) had a noteworthy impact on the model's performance. Our work does not focus on child-directed speech but on the comparison of story data versus open-domain data. Our aim is to provide a more diverse set of evaluation criteria for linguistic competence while also investigating the associated computing efficiency of the explored small and curated dataset.

Inspired by Timiryasov & Tastet (2023), we focus on pre-training small Llama-based architecture models. These models have been showcased to achieve competitive performance (Timiryasov & Tastet, 2023), with a notable reduction in training speed compared to a GPT-based architecture used in the TinyStories dataset by Eldan & Li (2023)).

3 Experimental Method

We set up an experimental pipeline for pre-training, evaluating, and comparing variants of Llama-based transformer models with a decoder-only architecture. The aim is to derive insights into three main factors across these datasets: formal linguistic competence, functional linguistic competence, and computational efficiency.

Figure 1 depicts the experimental pipeline, which consists of the following steps:

[2] https://babylm.github.io/

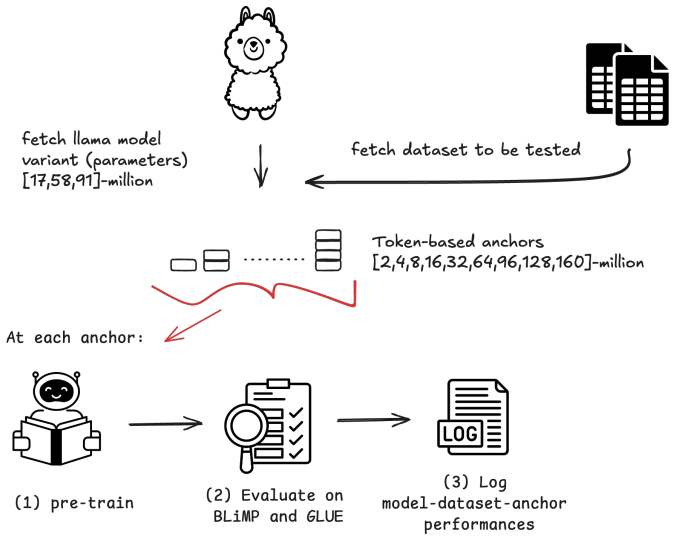

Figure 1: The experimental pipeline that we propose. For each model–dataset tuple, we pre-train the model on each anchor of the dataset (from scratch), evaluate on BLiMP and a subset of GLUE, and finally log the performances. The logged performances are used to create plots (such as learning curves) and execute task-oriented performance analyses.

1. **Fetch Llama model variant**: we select a Llama-based model variant from a predefined set of configurations. Details can be found in Section 3.2.
2. **Fetch the dataset**: we retrieve the dataset to be tested, choosing between **BabyLM-**100**m** (Choshen et al., 2024) or **TinyStories** (Eldan & Li, 2023).
3. **Token-based anchors**: we create various subsamples of the dataset with an increasing number of tokens. We extract progressively larger portions of tokens, following the predefined anchor points: $[2, 4, 8, 16, 32, 64, 96, 128, 160]$ (times million). These anchor sizes were determined using a geometric sampling scheme (Provost et al., 1999), with as final anchor the full size of the smallest dataset, i.e., **BabyLM-**100**m** with $\sim$160 million tokens. Details of the datasets can be found in Section 3.1, with a general overview in Table 1.
4. **Pre-training at each anchor**: we pre-train the model at each anchor point from scratch. We repeat the pre-training three times for each anchor–model–dataset combination to mitigate the effect of randomness and observe variance induced in the pre-training phase.
5. **Evaluation on benchmark tasks**: after pre-training, we evaluate the models on two common benchmarks: **GLUE** and **BLiMP** (see Section 3.3). We fine-tune the pre-trained models individually on a subset of GLUE tasks.

3.1 Datasets

The two datasets investigated in this work are the **TinyStories** (Eldan & Li, 2023) and the 100-million variant of the 2024 **BabyLM** challenge (Choshen et al., 2024). For simplicity, the 100-million BabyLM dataset will be referred to as the **BabyLM** dataset hereafter.

The choice of datasets stems from the aim to investigate small and curated datasets, to support more efficient pre-training in resource-constrained settings. Additionally, they can potentially be used to retrieve an estimation of the ultimate performance that can be achieved by a candidate model, enabling researchers to drop unfavourable model architectures early. Both datasets simulate aspects inspired by human development, but they vary greatly in their domains and the underlying structure of the language within the corpus. The aim is to investigate how these differences impact the performance of models in linguistic competence tasks, as well as investigate their compute efficiency.

Pre-training and tokeniser settings: We use causal language modelling (i.e., next-token prediction) as pre-training paradigm. We selected the hyperparameters for pre-training based on existing

Dataset	Total words	Unique words	Tokeniser vocabulary size
BabyLM	∼100M	∼1.7M	16K
TinyStories	∼450M	∼250K	8K

Table 1: The total words, unique words, and tokeniser vocabulary sizes of the datasets.

literature and parameters used in earlier studies (Timiryasov & Tastet, 2023; Eldan & Li, 2023), as well as a small-scale pilot study. Prior work (Ali et al., 2024; Tao et al., 2024) examined the impact of tokeniser choice and vocabulary size. Based on small-scale experimentation and this literature, we choose the *GPT2–Tokenizer BPE-based*. It seems to work well with language models for the English language (Ali et al., 2024), and it should be able to better handle out-of-vocabulary words. The vocabulary size affects the balance between word representation and computation cost. A larger vocabulary improves the representation, reducing the chance of encountering out-of-vocabulary words, but increases memory and computation costs. Therefore, we considered both the total number of words and the count of unique words in each dataset to determine the appropriate tokeniser vocabulary size. The decision was guided by Zipf's law (Manning & Schütze, 1999), which outlines that there is a long tail distribution of word frequency.

Specifically, we allocated a smaller vocabulary size to the TinyStories dataset, where a vocabulary of 8K tokens should effectively represent its word distribution. In contrast, the BabyLM dataset, which has a higher ratio of unique words to total words, was provided with a slightly larger vocabulary of 16K tokens. Table 1 summarises the datasets and the chosen tokeniser vocabulary sizes.

3.2 MODELS

We use model variants of the Llama-based architecture (decoder-only transformer models). The model variants are distinguished by their total number of trainable parameters. To retrieve the variants of the model, we alter the configuration of the Llama model (through the Hugging Face interface (Wolf et al., 2020)), and adapt (i) the hidden size, which impacts capacity, (ii) the intermediate size, which impacts complexity, (iii) the number of heads, which impacts parallel focus, and (iv) the number of stacked transformer decoders, which defines model depth. The model sizes are: 17-million, 58-million, 91-million trainable parameters, and their configuration is provided in Appendix A. Below, model references adhere to the following naming scheme: 17M, 58M, and 91M, where M denotes millions of parameters. Although the choice of models has been inspired by Timiryasov & Tastet (2023), we exclude knowledge distillation to focus purely on comparing intrinsic model performance across datasets rather than optimising final performance.

Given three model variants, two datasets, ten anchors (including the full dataset), and three repetitions, we pre-train a total of 180 model instances. Each instance is then fine-tuned on the individual tasks of the GLUE subset.

3.3 EVALUATION

We compare and analyse the performance of the models pre-trained on both datasets in two aspects: the evaluation of the linguistic competence achieved by models trained on said datasets and their associated compute efficiency. For the evaluation of linguistic competence we use a modified version of the *language model evaluation harness* (Sutawika et al., 2024; Choshen et al., 2024).

1. Linguistic competence: We follow Mahowald et al. (2023)'s distinction between *formal* and *functional* competence.

Formal linguistic competence refers to the ability to distinguish between grammatically correct and incorrect formation of a language. To evaluate this, we use the **BLiMP** benchmark, where minimal sentence pairs are provided to the pre-trained model in order to retrieve probabilistic assignments of the correct variant. This is done by averaging the log probabilities of the tokens in each sentence, as predicted by the model. No specific fine-tuning is required for evaluating on BLiMP.

Functional linguistic competence, also known as natural language understanding (NLU), refers to the ability to understand language in ways necessary for using it in real-world contexts. **GLUE**

evaluates this by using a collection of downstream tasks that require various forms of NLU. Each model has undergone a specific fine-tuning process for each of the selected downstream tasks.

2. Compute efficiency: is estimated using a token-based measure, where models are incrementally trained on larger token subsets of the dataset and evaluated at fixed anchor points. These anchors provide a consistent reference for assessing and comparing performance during training.

The selection of tokens as a proxy of compute requirements is based on the proportional relationship between training compute and tokens. The computational cost for training transformer-like models follows the approximation (Austin et al., 2025):

$$\text{Training Floating-point operations per second (FLOPS)} \approx 6 \times \text{params} \times \text{tokens}$$

where `params` is the number of trainable parameters, and `tokens` represent the training dataset size (in terms of tokens). Further evidence of this linear relationship is provided by Hoffmann et al. (2022). Since we compare models of the same size and use tokens as anchors, we retain the arguments of the formula constant at each anchor.

4 RESULTS

In this section, we discuss the results of training and evaluating models of three different sizes on the two datasets in varying dataset sizes. Conclusions on compute efficiency are incorporated in the subsections and results of formal and functional linguistic competence.

4.1 FORMAL LINGUISTIC COMPETENCE

Following our experimental pipeline (see Figure 1), we first evaluate formal linguistic competence.

Effect of model and dataset size: Figure 2 shows learning curves of the performance of three model sizes (17M, 58M and 91M) trained on the BabyLM and TinyStories datasets. At each anchor point, the models are pre-trained from scratch using the indicated number of tokens from the respective datasets. The trained models are then evaluated on their formal linguistic competence using the BLiMP benchmark. Each point on the learning curve indicates the average performance.

Comparing the two datasets indicates that models trained with the BabyLM dataset achieve higher overall BLiMP accuracy across the whole learning curve. Moreover, the models trained on TinyStories show slower improvement and reach a performance plateau at around 64 million sampled tokens for all model sizes. Additionally, results on TinyStories showcase greater variability and are more prone to induced randomness. This observation suggests that models trained on the TinyStories dataset are more sensitive to initial conditions of model parameters and random factors during the pre-training phase. It could also imply a higher likelihood of unstable convergence.

Furthermore, the results indicate that the size of the model has a substantial impact on performance, with the 91M model consistently outperforming the 58M model across all anchor points. This is particularly evident for the BabyLM dataset, where the performance gap across model sizes is more substantial. While this result is to be expected, this adds credibility to the experimental setup.

Finally, we can observe the ultimate dataset performance (i.e., using the full dataset) by investigating the dashed lines in Figure 2. Analysis of this is particularly important in the case of the TinyStories dataset, given that only a fraction of the $\sim$400-million token dataset had been considered up to 160 million tokens, whereas the BabyLM dataset is almost fully used. The lack of significant improvement on the full dataset suggests that additional data do not reveal further properties of language structure that could benefit the training of the model.

Possibly, the TinyStories dataset lacks rich linguistic features due to its vocabulary that was kept simple by design; after all, for the creation of the dataset, GPT-3.5 and GPT-4 were prompted to generate stories using only words that can be typically understood by children aged 3–4. In addition, since the dataset is synthetic, there is little variation in the structure of the stories. These factors could likely explain the inability of models trained on TinyStories to reach higher accuracy on BLiMP, with clear stagnation around the 64 million tokens.

Evaluation of individual tasks: To extract further insights on the performance of model variants across the datasets, we plot a heatmap to display the performance across the individual BLiMP tasks,

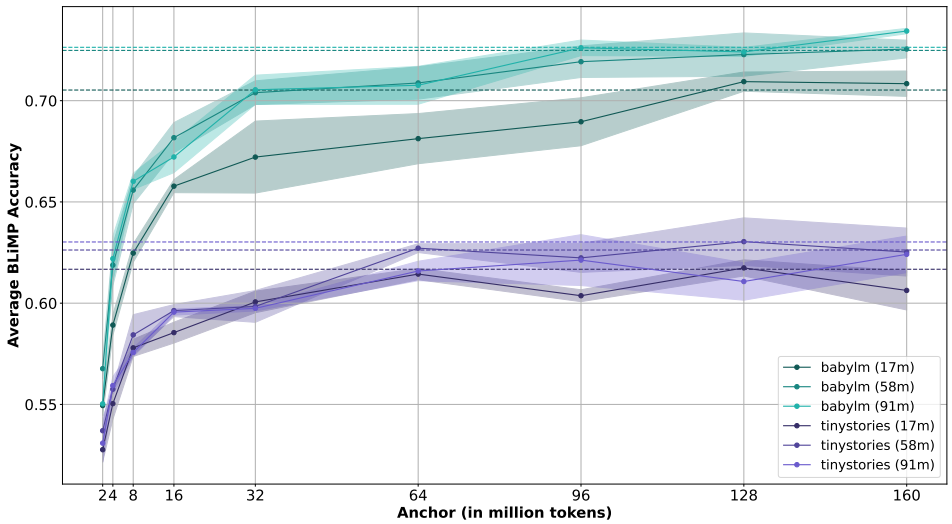

Figure 2: Learning curves (on three repetitions) for BLiMP performance comparing model sizes (17M, 58M and 91M) on two datasets: **BabyLM** and **TinyStories**. Dashed lines represent the performance on the full dataset for each model-dataset combination, following the same colouration.

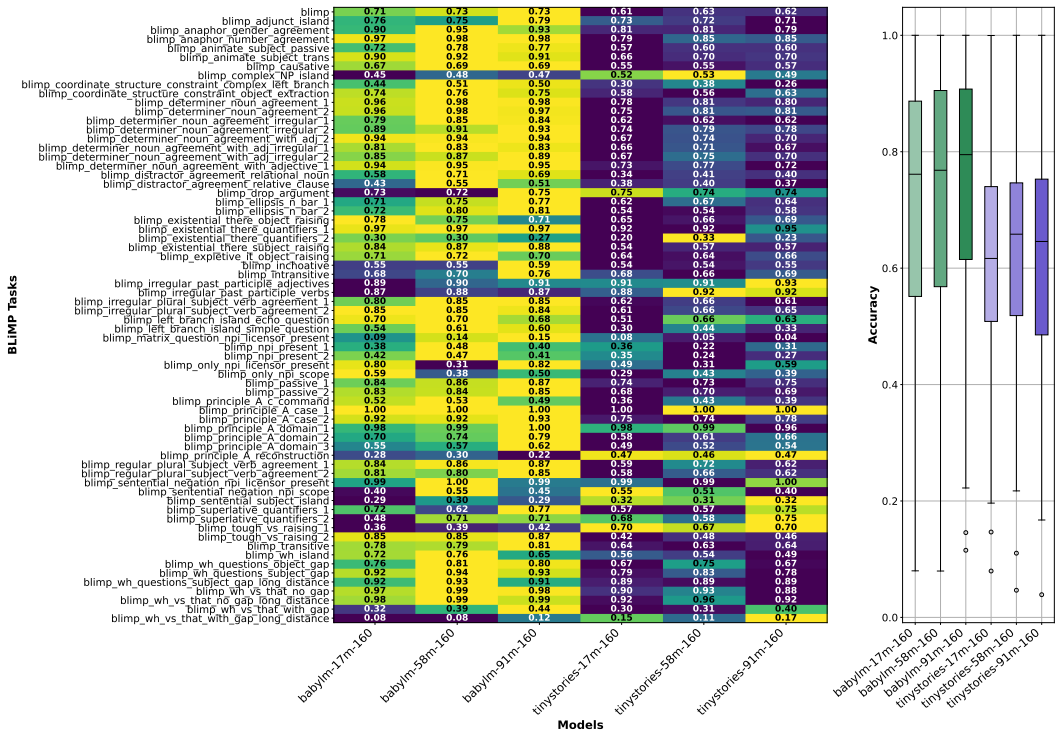

Figure 3: Heatmap of average accuracy scores (for three repetitions) across models (17M, 58M, and 91M) and all BLiMP tasks on two datasets: BabyLM and TinyStories, with 160M tokens (this being the largest anchor that occurs in both datasets). Each cell reports the score of a given fine-tuned model on a specific task. The background colour gives a row-wise indication of how the specific model performed, with yellow colours indicating better performance. The boxplot on the right depicts the distributions of performances across the BLiMP tasks.

along with a boxplot to showcase the distribution of performance across the tasks. For this, we used the models pre-trained on the final anchor of the dataset (i.e., the largest portion of the data). These

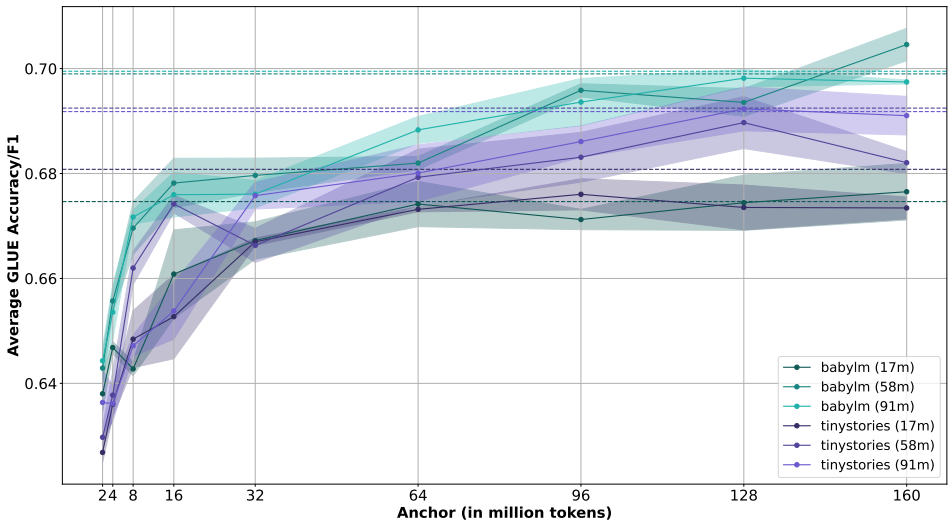

Figure 4: Learning curves (on three repetitions) for a subset of GLUE tasks comparing model sizes (17m, 58M and 91M) on two datasets: BabyLM and TinyStories. Dashed lines represent the performance on the full dataset for each model-dataset combination, following the same colouration.

can be observed in Figure 3. The yellow cell represents the best-retrieved performance on a task (i.e., in a row), while the darkest blue represents the worst-performing model. Moreover, the scores on the task are included in each cell.

Our findings show substantial variations in task allocation and overall performance and highlight once more the influence of model size. Upon first examining the heatmap (located on the left side of Figure 3), it is clear that models trained on BabyLM give better and more consistent performance across tasks than models trained on TinyStories. Results from larger model sizes are notably better; for the BabyLM dataset, this improvement is more noticeable across model sizes.

Analysis of the distribution via the boxplot (see right side of Figure 3) reveals tighter distributions and higher medians for models trained on BabyLM. This indicates more consistent task competence. In contrast, models trained on TinyStories exhibit greater variability and more extreme outliers, suggesting dataset-specific challenges. Finally, scaling the model size improves the performance for both datasets, with greater improvements for models trained on BabyLM. Increasing the model size clearly improves the median accuracy (reflected by the rise of the central line in the boxplot). Additionally, the boxplot constricts, indicating a reduction in variability of performance across the tasks, which highlights the benefit of scaling model capacity.

4.2 Functional Linguistic Competence

To evaluate the functional linguistic competence of the trained models, we use a collection of downstream tasks extracted from the GLUE benchmark suite. Figures 4 and 5 present the performance metric scores obtained during the training phase, focusing on validation scores.

Effect of dataset size: Figure 4 shows the learning curves. Each point depicts the average GLUE performance (across the repetitions) for each of the three model sizes (17M, 58M and 91M) pre-trained on the BabyLM and TinyStories datasets. At each anchor point, candidate models are pre-trained on sampled tokens from the respective datasets. Next, they are fine-tuned using predetermined hyperparameters following Timiryasov & Tastet (2023). We utilise the following tasks of the GLUE benchmark: CoLA, MNLI-mm, MRPC, QNLI, QQP, RTE, and SST-2.

Similar to the BLiMP evaluation (see Section 4.1), model–variants of all sizes perform better when trained on the BabyLM dataset compared to the TinyStories. The performance gap here is less pronounced but still evident. It can also be observed that the training of TinyStories across the anchor points presents greater fluctuations and higher variance across the repetitions. The 17M model stagnates around 64M tokens for both datasets. However, unlike the BLiMP results, larger

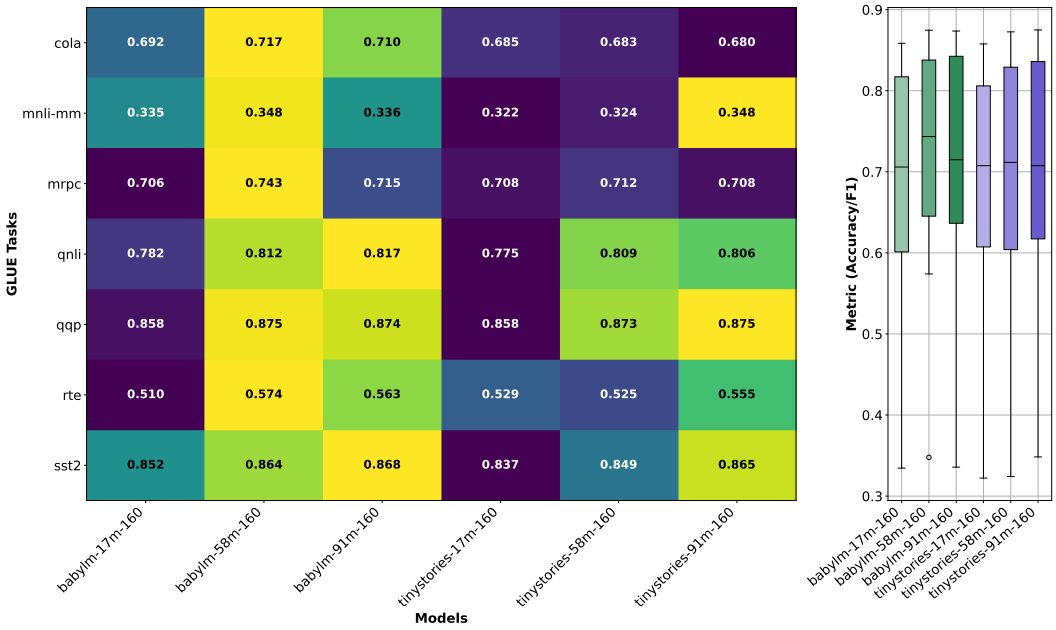

Figure 5: Heatmap of average accuracy scores (for three repetitions) across models (17M, 58M, and 91M) and GLUE tasks on two datasets: BabyLM and TinyStories, with 160M tokens (this being the largest anchor that occurs in both datasets). Each cell reports the score of a given fine-tuned model on a specific task. The background colour gives a row-wise indication of how the specific model performed, with yellow colours indicating better performance. The boxplot on the right depicts the distributions of performances across the GLUE tasks.

models (58M and 91M) consistently improve across all anchors on both datasets, reflecting their ability to capture more information with increased data.

Notably, unlike for the BLiMP results, using the full TinyStories dataset beyond 160M tokens (dashed lines in Figure 4) substantially improves performance. This suggests that the underlying data in TinyStories, despite being limited in vocabulary and complex grammatical structure, does enable the models to effectively capture fundamental linguistic patterns and structures needed for language *use* and *understanding*. Increasing the number of data samples boosts downstream task performance, indicating that data scale is important here.

Evaluation of individual tasks: Figure 5 shows the performance across individual tasks of the GLUE benchmark and their distribution. As previously, we used the models pre-trained on the final anchor of the dataset. Similarly to the previous heatmap, the scores per task are displayed, and the colour indicates the performance of a model compared to other models on that task (i.e., row). Even though the different datasets conclude on similar performances, BabyLM still outperforms the TinyStories dataset. We see that the BabyLM trained models have overall better performance across the different tasks (i.e., rows in the heatmap).

When looking into the distribution of scores (boxplot in Figure 5), it can be observed that moving from the 17M model to the 58M model variant presents a higher performance boost when trained on the BabyLM dataset. This can be inferred from the higher median score as well as constricted bounds, showcasing less variance. Models trained on TinyStories show a slightly higher median score with no substantial change in the outliers. Analysis of the 91M model for both datasets shows small to no improvement with the BabyLM pre-trained variant receiving lower average score (see Figure 4) and lower median scores as well as higher variance of scores across tasks (see Figure 5). This can be partially explained by the lack of hyperparameter optimisation across the model-size variants; current parameter settings were optimised in prior literature for the 58M model (Timiryasov & Tastet, 2023). The model pre-trained on TinyStories still shows minor improvements.

5 LIMITATIONS AND FUTURE WORK

Our work has revealed several valuable insights, but it also leaves room for further extensions and improvements. For instance, exploring the TinyDialogues dataset (Feng et al., 2024) could provide insights into how child-directed speech compares to the datasets analysed here in terms of linguistic competence and computational efficiency.

Another possible scope extension is to experiment with a greater variety of model architectures and sizes. In the current version, we have limited ourselves to Llama-based variants of relatively small size and with a static tokeniser per dataset, but this could clearly be extended in various directions, such as other model architectures, larger model sizes, tokeniser vocabulary sizes and other training paradigms (e.g., model distillation and reinforcement learning (DeepSeek-AI et al., 2025)).

The subset of the GLUE benchmark that we used includes a variety of downstream tasks, but integrating the full benchmark or tasks from more diverse benchmarking suites (e.g., SuperGLUE (Wang et al., 2019a)) could offer a more comprehensive and concrete assessment of functional linguistic competence.

Finally, the use of hyperparameter optimisation can further improve the downstream task performance (Baratchi et al., 2024). Tornede et al. (2024) speculate that the fine-tuning stage can be optimised while optimising all stages of the training pipeline is too costly. In this work, we adopted hyperparameter settings from (Timiryasov & Tastet, 2023), and did not tailor these to the various model sizes and tasks. Optimising the hyperparameters to all specific downstream tasks would provide more reliable results for the ultimate performance of the various models.

6 CONCLUSIONS

In this work, we have assessed the linguistic competence and computational efficiency of generative language models that were trained on small, curated datasets. We have analysed the learning curves with increasing amounts of data in the pre-training stage using two datasets that were originally designed with constraints inspired by human development. We have evaluated linguistic competence in two dimensions: (i) formal linguistic competence and (ii) functional linguistic competence. We have estimated computational efficiency with token-based measures, considering progressively larger dataset samples (anchors) that lead to a learning curve.

Our findings indicate that models trained with the **BabyLM** dataset outperform models trained with the TinyStories dataset on *formal* linguistic competence. The gap in *functional* linguistic competence is less pronounced, with models trained on **TinyStories** achieving comparable performance but showing smaller deviations across model sizes. Additionally, pre-training with **BabyLM** yields more consistent improvements across different dataset sizes, with lower variance across repetitions. These empirical findings suggest that **TinyStories** lacks certain rich linguistic features, likely due to its simpler vocabulary and synthetic origin, yet still supports the idea that narrative structures could contribute to functional linguistic competence.

These insights support the usage of **BabyLM** for pre-training language models in resource-constrained settings, where performance on small dataset samples can reliably predict the final achieved performance (such as mixture of experts (MoE) development). Moreover, the results seem more robust to variations of random seeds, as demonstrated by the lower standard deviations of results). On the other hand, the results of GLUE downstream tasks of models pre-trained **TinyStories** demonstrate the potential of narrative data for pre-training. Future work could consider the application of curriculum learning, where progressively complex data could be incorporated to enrich vocabulary. Such data could be sourced from other domains or more complex narrative samples. Finally, extending the datasets included and the set of downstream tasks considered will allow us to assess the generalizability of the insights we have obtained.

ACKNOWLEDGMENTS

This research is part of the project LESSEN with project number NWA.1389.20.183 of the research program NWA-ORC 2020/21, which is (partly) funded by the Dutch Research Council (NWO).

REFERENCES

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Schulze Buschhoff, Charvi Jain, Alexander Arno Weber, Lena Jurkschat, Hammam Abdelwahab, Chelsea John, Pedro Ortiz Suarez, Malte Ostendorff, Samuel Weinbach, Rafet Sifa, Stefan Kesselheim, and Nicolas Flores-Herr. Tokenizer choice for LLM training: Negligible or crucial? In *Findings of the Association for Computational Linguistics, NAACL 2024*, pp. 3907–3924, 2024.

Jacob Austin, Sholto Douglas, Roy Frostig, Anselm Levskaya, Charlie Chen, Sharad Vikram, Federico Lebron, Peter Choy, Vinay Ramasesh, Albert Webson, and Reiner Pope. How to scale your model. `https://jax-ml.github.io/scaling-book/`, 2025. Online book, retrieved: April 14th, 2025.

Mitra Baratchi, Can Wang, Steffen Limmer, Jan Nicolaas van Rijn, Holger Hoos, Thomas Bäck, and Markus Olhofer. Automated machine learning: past, present and future. *Artificial Intelligence Review*, 57, 2024.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. Guiding LLMs the right way: Fast, non-invasive constrained generation. *Computing Research Repository, CoRR*, abs/2403.06988, 2024.

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. The 2nd BabyLM challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Computing Research Repository, CoRR*, abs/2404.06214, 2024.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Computing Research Repository, CoRR*, abs/2501.12948, 2025.

Payal Dhar. The carbon impact of artificial intelligence. *Nature Machine Intelligence*, 2:423–425, 2020.

Ronen Eldan and Yuanzhi Li. TinyStories: How small can language models be and still speak coherent English? *Computing Research Repository, CoRR*, abs/2305.07759, 2023.

Steven Y. Feng, Noah D. Goodman, and Michael Frank. Is child-directed speech effective training data for language models? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2024*, pp. 22055–22071, 2024.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *Computing Research Repository, CoRR*, abs/2312.00752, 2023.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pp. 8342–8360, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *Computing Research Repository, CoRR*, abs/2203.15556, 2022.

Philip A. Huebner, Elior Sulem, Cynthia Fisher, and Dan Roth. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the Conference on Computational Natural Language Learning, CoNLL 2021*, pp. 624–646, 2021.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, ACL and International Joint Conference on Natural Language Processing (Volume 1: Long Papers), IJCNLP, 2021*, pp. 4582–4597, 2021.

Pierre Lison and Jörg Tiedemann. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, and Stelios Piperidis (eds.), *Proceedings of International Conference on Language Resources and Evaluation, LREC 2016*, 2016.

Brian MacWhinney. *The CHILDES Project: Tools for Analyzing Talk*. Lawrence Erlbaum Associates, third edition, 2000.

Kyle Mahowald, Anna A. Ivanova, Idan Asher Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. Dissociating language and thought in large language models: a cognitive perspective. *Computing Research Repository, CoRR*, abs/2301.06627, 2023.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

Felix Mohr and Jan N. van Rijn. Learning curves for decision making in supervised machine learning: a survey. *Machine Learning*, 113(11):8371–8425, 2024.

Foster J. Provost, David D. Jensen, and Tim Oates. Efficient progressive sampling. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 23–32. ACM, 1999.

Lintang Sutawika, Hailey Schoelkopf, Leo Gao, Baber Abbasi, Stella Biderman, Jonathan Tow, ben fattori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Aflah, Niklas Muennighoff, Thomas Wang, sdtblck, nopperl, gakada, tttyuntian, researcher2, Julen Etxaniz, Chris, Hanwool Albert Lee, Zdeněk Kasner, Khalid, LSinev, Jeffrey Hsu, Anjor Kanekar, KonradSzafer, and AndyZwei. EleutherAI/lm-evaluation-harness: v0.4.3. `https://doi.org/10.5281/zenodo.12608602`, 2024.

Ilya Sutskever. Pre-training as we know it will end. Invited Talk at Annual Conference on Neural Information Processing Systems, NeurIPS 2024, 2024.

Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. Scaling laws with vocabulary: Larger models deserve larger vocabularies. *Computing Research Repository, CoRR*, abs/2407.13623, 2024.

Inar Timiryasov and Jean-Loup Tastet. Baby Llama: Knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. *Computing Research Repository, CoRR*, abs/2308.02019, 2023.

Alexander Tornede, Difan Deng, Theresa Eimer, Joseph Giovanelli, Aditya Mohan, Tim Ruhkopf, Sarah Segel, Daphne Theodorakopoulos, Tanja Tornede, Henning Wachsmuth, and Marius Lindauer. AutoML in the age of large language models: Current challenges, future opportunities and risks. *Transactions on Machine Learning Research*, 2024, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *In Advances in Neural Information Processing Systems 30, NeurIPS 2017*, pp. 6000–6010, 2017.

Suzan Verberne. Is the search engine of the future a chatbot?, 2024. Inaugural lecture, Leiden University, 3 June 2024.

Tom J. Viering and Marco Loog. The shape of learning curves: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7799–7819, 2023.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *In Advances in Neural Information Processing Systems 32, NeurIPS 2019*, pp. 3261–3275, 2019a.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the International Conference on Learning Representations, ICLR 2019*, 2019b.

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

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics TACL*, 8:377–392, 2020.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell (eds.). *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, Singapore, 2023. Association for Computational Linguistics. URL `https://aclanthology.org/2023.conll-babylm.0/`.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022, 2022.

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, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020*, pp. 38–45, 2020.

Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the International Conference on Machine Learning, ICML 2023*, volume 202, pp. 38087–38099, 2023.

Yumo Xu and Mirella Lapata. Weakly supervised domain detection. *Transactions of the Association for Computational Linguistics TACL*, 7:581–596, 2019.

A MODEL SIZE VARIANTS

Table 2 depicts the different Llama-based decoder-only transformer models, according their respective model-sizes and associated configuration settings.

Model size	Hidden size	Intermediate size	Attention heads	Transformer layers
17M	256	1024	8	8
58M	512	1024	8	16
91M	768	2048	12	10

Table 2: Llama Model Variants: Configurations explored across different model sizes, depth and breadth. The models include 17-million, 58-million, 91-million parameters.