CoNLL 2024

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning

Proceedings of the Second BabyLM Challenge

November 15-16, 2024

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-222-0

Introduction

We are excited to welcome the second batch of babies to the 2024 BabyLM Challenge. The session will be held at the Conference on Computational Natural Language Learning (CoNLL; colocated with EMNLP) on November 15, 2024 in Miami, Florida, USA.

This year, the program includes an oral session for the award-winning papers, and a poster session for all accepted submissions. There is also an introductory presentation from the organizers summarizing the challenge, this year's winning submissions, and trends across submissions.

We received 31 submissions this year. A research challenge of this scale requires the participation of many parties, and we extend a big thank you to all of them. The participants' efforts are essential to advancing the state of cognitively plausible and sample-efficient language modeling—and to democratizing language modeling research.

We also extend our sincere thanks to the CoNLL organizers for hosting the BabyLM Challenge for the second time this year. Their efforts to integrate the proceedings of this challenge have been significant, and have in turn given a home to the significant efforts of the participants.

Finally, we thank our program committee members—largely sampled from the participants—for committing their time to help us curate an excellent program.

—The BabyLM Organizing Committee: Michael Y. Hu, Aaron Mueller, Candace Ross,
Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Alex Warstadt, Ethan Wilcox

Organizing Committee

Organizing Committee

Michael Y. Hu, New York University
Aaron Mueller, Northeastern University and Technion – Israel Institute of Technology
Candace Ross, Meta
Adina Williams, FAIR (Meta Platforms Inc.)
Tal Linzen, New York University and Google
Chengxu Zhuang, Massachusetts Institute of Technology
Ryan Cotterell, ETH Zürich
Alex Warstadt, ETH Zürich
Leshem Choshen, Massachusetts Institute of Technology and International Business Machines
Ethan Gotlieb Wilcox, Georgetown University

Table of Contents

Findings of the Second BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora
Michael Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt and Ethan Gotlieb Wilcox 1

Towards Data-Efficient Language Models: A Child-Inspired Approach to Language Learning
Mohammad Amin Ghanizadeh and Mohammad Javad Dousti 22

BabyLM Challenge: Experimenting with Self-Distillation and Reverse-Distillation for Language Model Pre-Training on Constrained Datasets
Aakarsh Nair, Alina Hancharova, Mayank Kumar and Ali Gharaee 28

From Babble to Words: Pre-Training Language Models on Continuous Streams of Phonemes
Zebulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery and Lisa Beinborn . . 37

Graphemes vs. phonemes: battling it out in character-based language models
Bastian Bunzeck, Daniel Duran, Leonie Schade and Sina Zarrieß 54

Exploring Curriculum Learning for Vision-Language Tasks: A Study on Small-Scale Multimodal Training
Rohan Saha, Abrar Fahim, Alona Fyshe and Alex Murphy 65

BabyHGRN: Exploring RNNs for Sample-Efficient Language Modeling
Patrick Haller, Jonas Golde and Alan Akbik 82

Choosy Babies Need One Coach: Inducing Mode-Seeking Behavior in BabyLlama with Reverse KL Divergence
Shi Shaozhen, Yevgen Matusevych and Malvina Nissim 95

Different Ways to Forget: Linguistic Gates in Recurrent Neural Networks
Cristiano Chesi, Matilde Barbini, Maria Letizia Piccini Bianchessi, Veronica Bressan, Achille Fusco, Sofia Neri, Sarah Rossi and Tommaso Sgrizzi 106

Developmentally Plausible Multimodal Language Models Are Highly Modular
Alina Klerings, Christian Bartelt and Aaron Mueller 118

ELC-ParserBERT: Low-Resource Language Modeling Utilizing a Parser Network With ELC-BERT
Rufus Behr 140

Extending the BabyLM Initiative : Promoting Diversity in Datasets and Metrics through High-Quality Linguistic Corpora
Laurent Prevot, Sheng-Fu Wang, Jou-An Chi and Shu-Kai Hsieh 147

Integrating Quasi-symbolic Conceptual Knowledge into Language Model Pre-training
Gábor Berend 159

Are BabyLMs Second Language Learners?
Lukas Edman, Lisa Bylinina, Faeze Ghorbanpour and Alexander Fraser 166

Less is More: Pre-Training Cross-Lingual Small-Scale Language Models with Cognitively-Plausible Curriculum Learning Strategies
Suchir Salhan, Richard Diehl Martinez, Zebulon Goriely and Paula Buttery 174

ConcreteGPT: A Baby GPT-2 Based on Lexical Concreteness and Curriculum Learning
Luca Capone, Alessandro Bondielli and Alessandro Lenci 189

When Babies Teach Babies: Can student knowledge sharing outperform Teacher-Guided Distillation on small datasets?
Srikrishna Iyer 197

Automatic Quality Estimation for Data Selection and Curriculum Learning
Hiep Nguyen, Lynn Yip and Justin DeBenedetto 212

Using Curriculum Masking Based on Child Language Development to Train a Large Language Model with Limited Training Data
Evan Lucas, Dylan Gaines, Tagore Rao Kosireddy, Kevin Li and Timothy Havens 221

WhatIf: Leveraging Word Vectors for Small-Scale Data Augmentation
Alex Mark Lyman and Bryce Hepner 229

A surprisal oracle for when every layer counts
Xudong Hong, Sharid Loáiciga and Asad B. Sayeed 237

Dreaming Out Loud: A Self-Synthesis Approach For Training Vision-Language Models With Developmentally Plausible Data
Badr AlKhamissi, Yingtian Tang, Abdulkadir Gokce, Johannes Mehrer and Martin Schrimpf 244

BabyLM Challenge: Exploring the effect of variation sets on language model training efficiency
Akari Haga, Akiyo Fukatsu, Miyu Oba, Arianna Bisazza and Yohei Oseki 252

BERT or GPT: why not both?
Lucas Georges Gabriel Charpentier and David Samuel 262

What should Baby Models read? Exploring Sample-Efficient Data Composition on Model Performance
Hong Meng Yam and Nathan Paek 284

BabyLlama-2: Ensemble-Distilled Models Consistently Outperform Teachers With Limited Data
Jean-Loup Tastet and Inar Timiryasov 292

Teaching Tiny Minds: Exploring Methods to Enhance Knowledge Distillation for Small Language Models
Hong Meng Yam and Nathan Paek 302

BERTtime Stories: Investigating the Role of Synthetic Story Data in Language Pre-training
Nikitas Theodoropoulos, Giorgos Filandrianos, Vassilis Lyberatos, Maria Lymperaiou and Giorgos Stamou 308

AntLM: Bridging Causal and Masked Language Models
Xinru Yu, Bin Guo, Shiwei Luo, Jie Wang, Tao Ji and Yuanbin Wu 324

Findings of the Second BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora

Michael Y. Hu[1] **Aaron Mueller**[2,3] **Candace Ross**[4]
Adina Williams[4,7] **Tal Linzen**[1] **Chengxu Zhuang**[6] **Ryan Cotterell**[8]
Leshem Choshen[5,6] **Alex Warstadt**[8] **Ethan Gotlieb Wilcox**[9]
[1]New York University [2]Northeastern University [3]Technion [4]Meta AI (FAIR)
[5]IBM Research [6]MIT [7]ML Commons
[8]ETH Zürich [9]Georgetown University
michael.hu@nyu.edu

Abstract

The BabyLM Challenge is a community effort to close the data-efficiency gap between human and computational language learners. Participants compete to optimize language model training on a fixed language data budget of 100 million words or less. This year, we released improved text corpora, as well as a vision-and-language corpus to facilitate research into cognitively plausible *vision* language models. Submissions were compared on evaluation tasks targeting grammatical ability, (visual) question answering, pragmatic abilities, and grounding, among other abilities. Participants could submit to a 10M-word text-only track, a 100M-word text-only track, and/or a 100M-word and image multimodal track. From 31 submissions employing diverse methods, a hybrid causal-masked language model architecture outperformed other approaches. No submissions outperformed the baselines in the multimodal track. In follow-up analyses, we found a strong relationship between training FLOPs and average performance across tasks, and that the best-performing submissions proposed changes to the training data, training objective, and model architecture. This year's BabyLM Challenge shows that there is still significant room for innovation in this setting, in particular for image-text modeling, but community-driven research can yield actionable insights about effective strategies for small-scale language modeling.

1 Introduction

This paper describes the second BabyLM Challenge and its findings. The broader goals and motivation of the challenge have remained constant since the first iteration last year. At the heart of both this year's and last year's challenge is the observation that children are incredibly data-efficient language learners, whereas artificial neural-network-based language models are not. On the one hand, children are exposed to less than 100 million word tokens by the age of 13 (Gilkerson et al., 2017), at which point they have mastered their native language(s). On the other hand, today's ANN-based language models are trained on trillions of words—five to six orders of magnitude more than the typical human language learner. For a more in-depth discussion on the issue of data efficiency, see the findings of last year's challenge (Warstadt et al., 2023) as well as Wilcox et al. (2024), a position piece written by many of the challenge organizers.

The learning discrepancy between humans and models raises two important questions: First, how is it that humans are able to learn language so efficiently? And second, what insights from human language learning can be used to improve language models? It is our hope that by creating a platform for interested parties to experiment with data-limited and cognitively inspired language modeling, we can continue to make progress on these interrelated questions. In particular, our goal with BabyLM is to contribute to:

1. Building more cognitively and developmentally plausible models of human language acquisition and processing, which can be used for the scientific study of language.

2. Optimizing training pipelines prior to scaling, allowing for faster iteration on architectures and hyperparameters.

3. Enabling research on language model training to a wider group of interested researchers, beyond highly-funded industry labs.

The main difference between this year's and last year's challenge is twofold: First, this year we allowed participants to bring their own datasets, as long as they stayed within the 100 million word limit for our *Strict* track, or the 10 million word limit for our *Strict-Small* track. The motivation behind this decision is that pretraining data quality has been linked to large improvement gains in at-scale language models (Gunasekar et al., 2023),

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 1–21
November 15-16, 2024

so this year we allowed participants to improve the training data beyond the provided dataset, which was effectively a dataset baseline. Second, this year included a *Multimodal* track, in which participants trained on aligned text-image data, and tested their models in a novel text-image evaluation pipeline. Non-linguistic information, such as visual input, potentially plays a large role in child language acquisition. While visual input is not inherently necessary for successful language acquisition (for example, blind children learn language largely without issue), visual grounding has been linked to faster language learning (Pérez-Pereira and Castro, 1992; Campbell et al., 2024). Furthermore, visual grounding has long been hypothesized to aid word learning: children learn nouns more easily than verbs (Gentner, 1982; McDonough et al., 2011), arguably because the former are more easily linked to visual stimuli than the latter. Additionally, children learn concrete nouns easier than abstract nouns (Bergelson and Swingley, 2013). However, visual grounding also presents several challenges: Words may be time-delayed with respect to their referents, or one word may be uttered in a context with multiple competing possible referents. With this in mind, our hope was that the *Multimodal* track would help to explore the space of possible computational models for visual grounding during language acquisition.

Findings and takeaways. This year, we received 31 submissions from 17 different countries making diverse contributions. Examples included submissions proposing novel architectures, new training objectives, innovating on knowledge distillation methods, and proposing curriculum learning methods, among others. We conduct a meta-analysis of the results, which yields several concrete recommendations. The best-performing submissions constructed their own training datasets, proposed new model architecture, or new training objectives. Performance on the BabyLM evaluations also correlated strongly with training FLOPs, suggesting that high-compute training regimes still tend to reliably perform better, even in low-data settings. The BabyLM research community also showed growing attention to tokenization and multilingual language modeling, while maintaining interest in curriculum learning and applying linguistic biases to language models.

Our data (pretraining corpora and evaluation data; [link]), preprocessing code [link], baselines [link] and evaluation pipeline [link] are all publicly available. We also release the submitted models of those who agreed to release them, along with their hyperparameters and results [link]. The leaderboard may be found here [link].

2 Competition Details

Tracks. The second BabyLM Challenge included three competition tracks: ***Strict***, ***Strict-Small***, and ***Multimodal***. Additionally, we opened a standalone ***Paper*** track, accepting research related to cognitive modeling with language models or small-scale pretraining, similar to a workshop.

The *Strict* and *Strict-Small* tracks required that submissions be trained on 100M words or less and 10M words or less, respectively. These tracks no longer required that participants use the fixed dataset from last year's challenge, although we still provided an updated version of this dataset, described in Section 3. Models in these tracks were evaluated on language-only evaluation tasks.

In the *Multimodal* track, participants trained multimodal image-text models. Participants were allowed to use any model and training procedure they desired, as long as the model could assign (pseudo) log-likelihoods to strings of text, conditioned on an image. Again, participants were free to construct their own datasets, including unlimited visual inputs, as long as the text data was within a 100M word budget. To facilitate easier participation in this track, we released a suggested multimodal dataset that consisted of 50% text-only and 50% paired image-text data. Submissions to this track were evaluated on both language-only and additional multimodal tasks.

3 Pretraining Corpus

This year, we updated the text-only dataset from the previous competition and provided a novel image-text dataset for the *Multimodal* track. Data for both the text-only and multimodal datasets can be downloaded from `https://osf.io/ad7qg/`.

For the text-only dataset updates, we increased the proportion of child-oriented data (counting both transcribed speech and written data) to 70% up from 39% last year, and we increased transcribed speech data to 58% up from 55% last year. We have eliminated the Wikipedia portion of the data (except for Simple English Wikipedia) due to being the only non-spoken and non-child-level data, and we have eliminated the QED portion due to qual-

Dataset	Description	# Words (multimodal)	# Words (strict)	# Images
Localized Narratives[a]	Image Caption	27M	–	0.6M
Conceptual Captions 3M[b]	Image Caption	23M	–	2.3M
CHILDES[c]	Child-directed speech	14.5M	29M	–
British National Corpus (BNC), dialogue portion[d]	Dialogue	4M	8M	–
Project Gutenberg (children's stories)[e]	Written English	13M	26M	–
OpenSubtitles[f]	Movie subtitles	10M	20M	–
Simple English Wikipedia[g]	Written Simple English	7.5M	15M	–
Switchboard Dialog Act Corpus[h]	Dialogue	0.5M	1M	–
Total	–	100M	100M	2.9M

Table 1: Datasets for the multimodal and strict tracks of the 2nd BabyLM Challenge. Word counts are approximate and subject to slight changes. [a]Pont-Tuset et al. (2020a) [b]Sharma et al. (2018a) [c]MacWhinney (2000) [d]Consortium (2007) [e]Gerlach and Font-Clos (2018) [f]Lison and Tiedemann (2016a) [g]`https://dumps.wikimedia.org/simplewiki/` [h]Stolcke et al. (2000)

ity issues. We have also reduced our reliance on OpenSubtitles, which can include scripted speech, which is arguably less ecologically valid than other spoken sources. CHILDES now comprises a significantly larger portion of the new dataset. We use the entire available English portion of CHILDES including both caregiver and child utterances, increasing the proportion of child-oriented discourse from 5% last year to 29%.[1] We also replaced last year's children's stories and Project Gutenberg data with a custom children's stories dataset sourced entirely from Project Gutenberg. We select child-appropriate books using the provided `subject` metadata, and then select the 1000 most common books, giving us a combined corpus of 26M words. For more details about other data sources, see (Warstadt et al., 2023).

In addition, we provide a novel image-text dataset to facilitate easier participation in the *Multimodal* track. This dataset has two components: First, we provide 50M words of text-only data, drawn from the 100M BabyLM corpus via stratified sampling (that is, we preserve the relative distribution from the different data sources). Second, we provide paired text-image data that includes 50M words of text. This paired data comes from two sources: 27M words from the Localized Narratives dataset (Pont-Tuset et al., 2020b) and 23M words from the Conceptual Captions 3M (CC3M) dataset (Sharma et al., 2018b). For the Localized Narratives dataset, we used the text captions and the images from the MS-COCO (Lin et al., 2014) and Open Images (Kuznetsova et al., 2020) subsets. For the CC3M dataset, we used the image-caption pairs whose images were still valid in January 2024. In the OSF directory at the above link, we provided scripts to download the images. Table 1 gives an overview of the datasets comprising the BabyLM pretraining set, and descriptions of each data source are provided in Appendix A.

3.1 Preprocessing

We released train, validation, and test splits for each of the ten data sources in *Strict* and *Strict-Small* in proportions 83.3%/8.3%/8.3%, respectively. The 10M word *Strict-Small* training set is sampled randomly from the *Strict* training set: after preprocessing, we downsampled and split each source by randomly sampling chunks of 2000 lines or longer. The code and instructions for downloading and preprocessing the raw data are publicly available.[2]

We performed minimal preprocessing in terms of filtering and reformatting text. Notably, we preserved newlines, meaning newlines do not consistently delimit documents, paragraphs, or sentences, as in some pretraining datasets. We used WikiExtractor (Attardi, 2015) to extract text from the `xml` Simple English Wikipedia dump dated 2022-12-01. We removed `<doc>` tags in Simple English Wikipedia and selected the spoken subset of the BNC by taking only lines from the `xml` containing the `<stext>` tag and extracting the text from the `xml`. We used code by Gerlach and Font-Clos (2020) to download and preprocess data from Project Gutenberg, which we additionally filtered to contain only English texts by authors born after 1850. The OpenSubtitles and Wikipedia portions of the pretraining corpus were shared with us in preprocessed form, having had duplicate documents

[1]We thank Brian MacWhinney (personal correspondence) for alerting us to the existence of this additional CHILDES data.

[2]`https://github.com/babylm/babylm_data_preprocessing`

removed from OpenSubtitles and preprocessing steps performed to Wikipedia similar to our Simple English Wikipedia procedure.[3] We used regular expressions to remove speaker and dialog act annotations from the Switchboard Dialog Act Corpus and annotations from the CHILDES data. We preserved speaker annotations and scene descriptions from CHILDES. We performed no preprocessing on the remaining datasets.

4 Evaluation and Submission

As in last year, we distributed a shared evaluation pipeline based on the LM Evaluation Harness (Gao et al., 2021). For the *Strict* and *Strict-Small* tracks, evaluation tasks were largely the same as the previous year: we used BLiMP (Warstadt et al., 2020), the BLiMP Supplement (Warstadt et al., 2023), and a subset of (Super)GLUE tasks (Wang et al., 2019, 2018a) as the public evaluation set. BLiMP measures whether LMs prefer grammatical to minimally-differing ungrammatical sentences (i.e., minimal pairs) and spans a range of grammatical phenomena including subject-verb agreement, binding, and control/raising constructions. The BLiMP supplement is a disjoint subset of minimal pairs designed specifically for last year's BabyLM Challenge to test linguistic knowledge not covered by BLiMP, such as dialogue and pragmatics. (Super)GLUE is designed to measure natural language understanding across a diverse array of subtasks; its tasks include question answering and natural language inference, among others.

For the *Multimodal* track, participants were required to evaluate on the evaluation benchmarks from the text tracks; this was to establish whether training on image data facilitated sample-efficient language modeling. In addition, we included a suite of multimodal evaluation tasks. The public evaluation datasets included Visual Question Answering (VQA; Antol et al., 2015; Goyal et al., 2017) and Winoground (Thrush et al., 2022). VQA measures whether vision-language models (VLMs) prefer correct answers to questions about visual inputs, and Winoground measures whether LMs prefer accurate descriptions of images among minimally differing options (e.g., given an image of dirt on top of a light bulb, does the VLM prefer "a lightbulb on top of dirt", or "dirt on top of a lightbulb", and vice versa given another image where the lightbulb is on top of dirt).

This year, we used the Elements of World Knowledge (EWoK) dataset (Ivanova et al., 2024) as the hidden task for the text tracks. This task measures pragmatic, commonsense, and discourse knowledge. For the *Multimodal* track, the hidden task was DevBench (Tan et al., 2024); this benchmark contains subtasks targeted at evaluating visual and linguistic abilities that emerge at different stages of children's development, including subtasks where (i) the model must pick the correct image associated with a given word; (ii) the model must pick the correct image corresponding to a sentence; and (iii) the model must assign appropriately higher or lower similarity scores to more or less similar images. The data for these tasks was released two weeks before the model submission deadline. We selected these tasks based on whether they capture distinct phenomena from the public evaluation tasks, such that optimizing only for individual tasks or narrow subsets of linguistic competencies would not be overly rewarded.

Most of the evaluation tasks were zero-shot. Zero-shot evaluation entails comparing the probabilities of different sequences of text. Thus, all submitted models were required to assign a (pseudo) log-likelihood to a sequence of tokens. Additionally, the (Super)GLUE tasks required fine-tuning a classification head appended to the model. Models did not need to *generate* sequences for any evaluation task; thus, both autoregressive and masked language modeling architectures could be used.

4.1 Evaluation Pipeline

We provided code to unify the evaluation setup across submissions. This was released as a public repository on GitHub.[4] The evaluation pipeline supports models implemented in HuggingFace, including Transformer-based architectures, structured state space models (e.g., Mamba; Gu and Dao, 2024), and recurrent neural networks (Peng et al., 2023), among other architectures. Note, however, that we did not restrict the model submissions to HuggingFace-based models; participants were allowed to use their own evaluation setup if desired, so long as they were able to produce predictions

[3] We thank Haau-Sing Li for allowing us to use this preprocessed data.

[4] https://github.com/babylm/evaluation-pipeline-2024

in the expected format.[5] For model and result submissions, users were required to (i) upload a link to their model (on any file-hosting service), and (ii) provide model predictions for each example of each task; we provided a template specifying the format of the predictions file in the evaluation pipeline repository.

Data preprocessing. NLP tasks in our evaluation pipeline often contained vocabulary that is not contained in the BabyLM pretraining corpora. To address this mismatch, we filtered each evaluation task according to its lexical content. We first computed two vocabularies by collecting all words that appear at least twice in the *Strict-Small* corpus and collecting all words that appear at least twice in the *Multimodal* corpus. Then, we took the intersection of these two vocabularies to obtain the final vocabulary. Finally, we iterated through each example in each evaluation task; if an example contained any words that appeared less than twice in the final vocabulary, we filtered the example. Otherwise, each dataset is presented in its original format. See Table 4 in Appendix B for details on the size of the filtered datasets.

4.1.1 Evaluation Paradigms

Zero-shot evaluation. For zero-shot tasks—all of them except (Super)GLUE—we modified the `lm-eval-harness` repository, originally by EleutherAI (Gao et al., 2021). This provides functionality for scoring autoregressive decoder-only LMs and encoder-decoder LMs. For encoder-only LMs, we modified the repository to support masked language model scoring as described in Salazar et al. (2020), and as updated by Kauf and Ivanova (2023).[6] We also modified the pipeline to support multimodal models and tasks.

Finetuning. Prior to the challenge, we experimented with zero-shot learning and few-shot in-context learning for (Super)GLUE. However, this often resulted in random-chance accuracies from our baselines; we therefore employed finetuning. While finetuning technically adds to the training set size, we consider this acceptable, as finetuning on a single GLUE or MSGS task does not meaningfully add to the domain-general linguistic abilities of language models. For tasks requiring finetuning—namely, (Super)GLUE (Wang et al., 2018b, 2019)—we base our scripts on HuggingFace's example finetuning scripts for text classification.[7] We modified the script from last year's pipeline to work with more recent versions of HuggingFace transformers. We provided a default set of hyperparameters that we found to work well across our baseline models, though participants were allowed to modify hyperparameters if they wished. We also provided support for fine-tuning models via low-rank adapters (LoRA; Hu et al., 2022). This enabled the possibility of faster and more compute-efficient model adaptation for our tasks.

4.2 Submission process

Submission format. The submission form was hosted via OpenReview. We required a link to the models, as well as a link to the predictions of these models for all examples for all tasks. The predictions file was formatted as a JSON; each example had an entry with an example ID as its key, and the the prediction of the model as its value. For classification tasks, a prediction was a label ID integer. For zero-shot tasks, predictions were the string that received the highest probability according to the model. The submission process for the competition consisted of three components, which are outlined below:

Paper submission. Each participant submitted a paper detailing their research, methodology, experimental design, and key findings. This was required for all participants, even if they did not submit a model to compete in the challenge.

Artifact submission. In addition to the paper, participants who opted to compete and adhere to the competition rules were required to provide supplementary materials, including model outputs, checkpoints, and pretraining data (unless the default pretraining dataset was used). Participants were also required to upload their predictions for all evaluation tasks.

Submission form. To facilitate comparability and reproducibility, participants were asked to fill in a standardized form that captured model metadata, including hyperparameters, submission de-

[5] Upon release of the evaluation pipeline, we announced that we would provide support as needed to teams training LMs not based in HuggingFace.

[6] We used the implementation of Misra (2022) in the `minicons` library.

[7] `https://github.com/huggingface/transformers/blob/211f93aab95d1c683494e61c3cf8ff10e1f5d6b7/examples/pytorch/text-classification/run_glue.py`

scriptions, and links to custom data if the standard corpus was not used.

4.3 Baselines

As opposed to last year's baselines, which were selected and trained relatively naively, this year's baselines were based on the architectures of winning submissions from last year's challenge. For the *Strict* and *Strict-Small* tracks, we released the following baselines: LTG-BERT (encoder-only; Samuel et al., 2023) and BabyLlama (decoder-only; Timiryasov and Tastet, 2023a). Although a variant of LTG-BERT (called ELC-BERT) won last year's challenge (Charpentier and Samuel, 2023), Wilcox et al. (2024) showed that similar performance on BabyLM evaluations can be achieved without the additional modifications of ELC-BERT. Thus, we chose LTG-BERT as the baseline, as it is a simpler model. BabyLlama is architecturally similar to Llama (albeit with far fewer parameters), and is additionally trained using knowledge distillation. For the *Multimodal* track, we released vision language models based on GIT (Wang et al., 2022) and Flamingo (Alayrac et al., 2022) architectures, both of which are autoregressive.

Implementation details. For LTG-BERT, we initially used the code provided in the repository linked in Samuel et al. (2023), but we encountered unstable training due to loss spikes with this setup. We therefore used the LTG-BERT model released on HuggingFace, and trained it using the HuggingFace trainer. While training was still relatively unstable compared to other architectures, this procedure yielded performance in the expected range relative to other baselines. For BabyLlama, we use the code from the repository linked in Timiryasov and Tastet (2023a), with small changes for compatibility with this year's BabyLM corpus. For the GIT and Flamingo baselines, we adapt the implementation of Zhuang et al. (2024). Note that these baselines are not necessarily meant to achieve high scores on our evaluation tasks; rather, they are meant to encourage participants to innovate and improve beyond naive applications of existing methods.

5 Competition Results

In this section, we discuss the overall results of the competition (§5.1), track winners (§5.2), and this year's Outstanding Papers (§5.3).

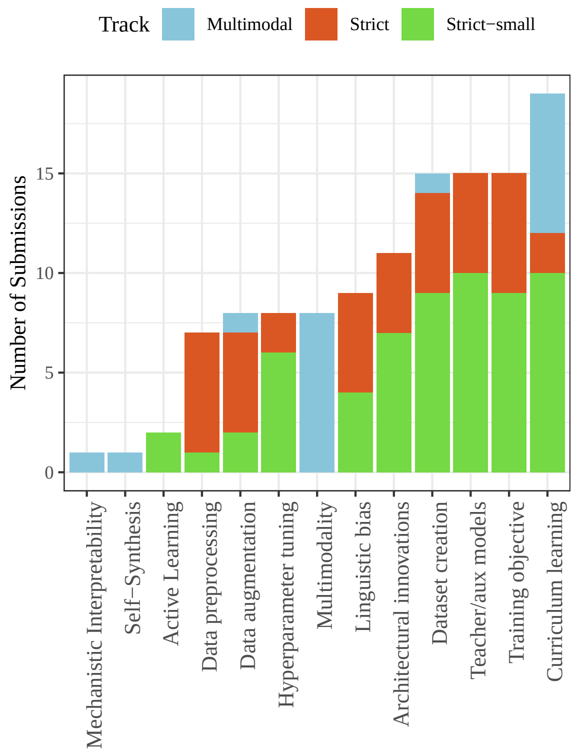


Figure 1: A breakdown of the various approaches used in the 2024 BabyLM challenge, organized by category and track. Curriculum learning again takes the top spot as the most popular approach, followed by training objective innovations.

We received 31 papers and 64 models in total, with two models submitted to the paper track. Table 2 shows the submission counts for each track. Despite efforts to make text–vision pretraining as accessible as possible, only three teams submitted to the *Multimodal* track, for a total of 8 model submissions. As none of these submissions outperformed our baselines, we decided not to award a winner in this track. Despite this disappointment, we hope that our datasets and evaluation resources serve as a basis for further exploration of text-image models in the years to come.

We found that many submissions focused their efforts on similar techniques. To better quantify this, we devised, in Figure 1, a typology of the most common approaches and assigned each submitted model one or more labels. §6.3 provides more detailed descriptions of each approach, as well as results indicating which ones were most effective.

All participants are affiliated with universities or independent research institutions. Participants' home institutions are located in 16 different countries. The number of participants by country is

	# Models	# Participants
Multimodal	8	3
Strict-Small	35	18
Strict	19	11
Total	*64*	*31*

Table 2: Total number of models and participants per track. Participants who submitted to multiple tracks are counted once in the total. Two models were submitted to the *Paper* track only.

as follows (multinational submissions are counted more than once): Germany (8), United States (6), Netherlands (4), Italy (2), UK (2), Canada (1), China (1), Greece (1), Hungary (1), Iran (1), Israel (1), Japan (1), Norway (1), Singapore (1), Sweden (1), Switzerland (1), and Taiwan (1).

5.1 Overall Results & Track Winners

The results from all submissions are shown in Figure 2, with the scores of the top-performing models in each track detailed in Table 3. In the figure, dashed gray lines show the performance of non-competition models (either baselines or skylines), and solid green lines show human performance on evaluation metrics. For GLUE, we use the human scores reported in Nangia and Bowman (2019) and for BLiMP we use the *individual* human agreement scores reported in Warstadt et al. (2020). For Winoground, we plot the human *group* score reported in Thrush et al. (2022), which is slightly more stringent than our model evaluation setup as it requires humans to make the correct judgments over a set of several comparisons. For VQA, we report the *Question + Image* score on *real* images reported in Antol et al. (2015). Again, the human task is arguably more difficult than our own evaluation as it assesses correctness in open-ended responses, rather than by comparing ground-truth captions to distractors. Therefore, the difference between the human and model scores on the vision tasks is likely an underestimate of the true difference between their respective visual capabilities.

We start our discussion by noting several high-level trends, before turning to the winning models. First, as with last year, we notice the same overall pattern of scores between our three different tracks—models in the *Strict* track tend to perform better than those in the *Strict-Small* (although the variance is higher), and models in the *Multimodal* track perform worse. *Ceteris paribus*, more data indeed helps models learn, and learning from multimodal data remains challenging. Within text evaluations, models also perform slightly better on BLiMP compared to GLUE, which is a trend we observed last year as well.

Did model performance improve over last year? At the upper end of the distribution, the answer is *yes*. This year, one model in the *Strict-Small* track beats our Llama skyline on BLiMP, and the best model in the *Strict* track is within just 2.5 percentage points shy of the human score on this task. In addition to these few high-performing models, we also observed a small upward shift in the distribution of model scores compared to last year. For example, last year only 5 models in the *Strict-Small* track achieved a GLUE score of higher than 70; this year that increased to 7 models. For the *Strict* track, this number was 7 last year and 8 this year. One explanation for this small upward shift is that this year we allowed contestants to bring their own data for the *Strict* and *Strict-Small* tracks, provided they stayed within the data limits for each track. Many contestants modified our provided data by procuring new sources, generating data from auxiliary language models, or filtering the existing data. As we shall see in section 6.3, dataset creation was an effective method, and we hypothesize that performance increases on our benchmark tasks over last year can be partially attributed to such data-related improvements.

The introduction of EWoK as our hidden evaluation allowed us to observe that current systems do not learn world knowledge within 100M words. Most submissions perform near chance, at 50% (where dots are colored purple); the maximum score was 58.4%.[8] This observation highlights a potential area for future research. It may be that the current BabyLM corpus—used by many of the submitting teams—simply does not contain the world knowledge that EWoK is designed to test. One other possibility is that existing architectures have a bias towards learning linguistic phenomena more

[8]Many masked language model submissions initially reported EWoK scores around 60–70%. This was likely due to a default behavior of the LM evaluation harness, which assigns a label of 0 when the probability of both sequences is the same. When changing this behavior to instead uniformly sample a label when the sequence probabilities are the same, most models get closer to 50–60% accuracy. We confirmed these scores using a scoring script not based in the LM evaluation harness. This only affected EWoK: we were able to closely reproduce the participant-submitted scores for all other zero-shot tasks, with or without uniform sampling.

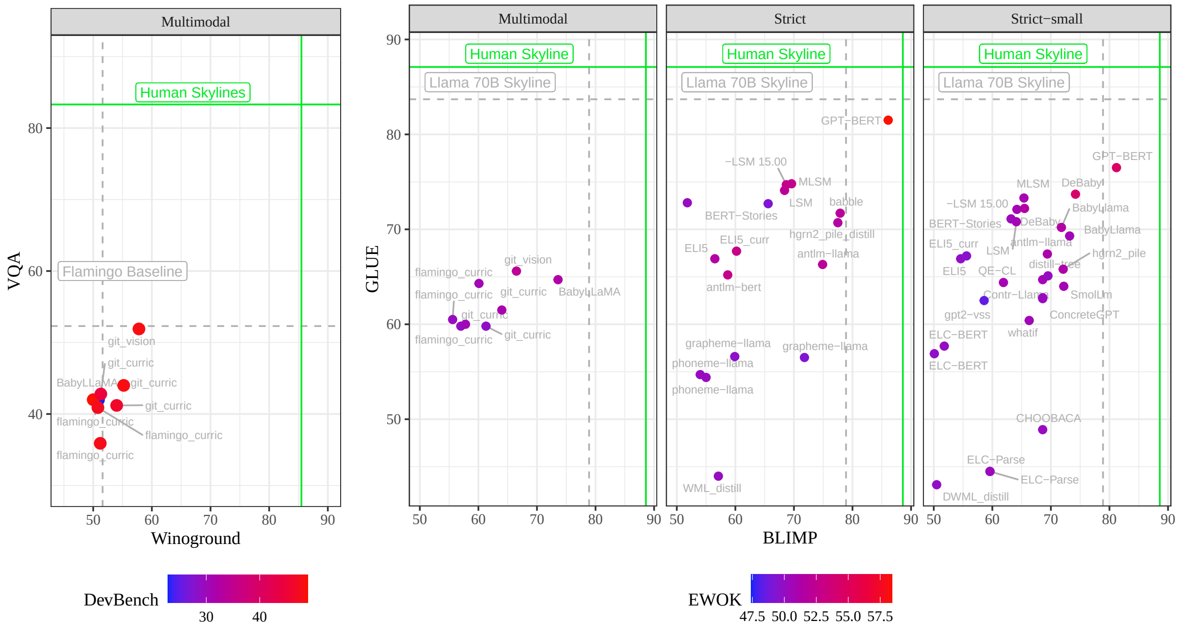


Figure 2: Overall results: At left, multimodal models on multimodal tasks; at right, all models on text tasks. N.B. Human scores for multimodal evals differ somewhat from how we evaluate our models.

easily than relationships between concepts, physical properties, and other topics covered by EWoK. Further work on data (perhaps including data attribution methods) and algorithms will help elucidate why EWoK is so challenging for BabyLM models.

Finally, the *Multimodal* track proved challenging, and no submission beat the baselines we released. We discuss this further in Section 5.2.

5.2 Winning Submissions

***Strict* and *Strict-Small* tracks.** The winner of both the *Strict* and *Strict-Small* tracks is GPT-BERT, submitted by (Charpentier and Samuel, 2024). GPT-BERT merges the causal (CLM) and masked language modeling (MLM) objectives from GPT and BERT, respectively, using the following key insight: by shifting MLM predictions one position to the right, the MLM predictions become aligned with next-token predictions from CLM. The authors use this insight to combine both objectives and seamlessly mix between MLM and CLM.

To train on MLM and CLM simultaneously, the authors duplicate the training data, masking and processing each copy differently for causal and masked language modeling. For each training batch, the authors choose to draw data from the CLM dataset copy with probability p and from the MLM dataset with probability $1 - p$. The authors explore a range of values for p, finding that a 1:7 causal-to-masked ratio tends to give good performance across a variety of tasks. GPT-BERT modifies the LTG-BERT architecture by adding gates on attention heads, as well as the residual connection reweighting proposed in ELC-BERT (Charpentier and Samuel, 2023), the winner of *Strict* and *Strict-Small* from last year.

A different submission to this year's competition, AntLM (Yu et al., 2024), also explored combining CLM and MLM by alternating between the two objectives on a per-epoch basis. The authors found that the best schedule for training LTG-BERT was 6 epochs of CLM, followed by 60 epochs of MLM, followed by 6 more epochs of CLM. While AntLM gets lower scores than GPT-BERT, it performs well overall, also beating our baselines. We conclude that 1) the LTG-BERT architecture remains a strong backbone for small language models, provided one can train it effectively, and 2) combining causal and masked language modeling objectives clearly improves performance over single objective baselines.

***Multimodal* track.** We did not award a winner for the *Multimodal* track this year. We received three submissions, and none outperformed the baselines we released. This speaks to the difficulty of multimodal learning in general. Leveraging both the text and vision modalities is challenging because the model can often learn unimodal shortcuts to

	Model	BLiMP	BLiMP Supplement	(Super)GLUE	EWoK	*Text Average*	VQA	Winoground	DevBench	*Vision Average*
Strict	GPT-BERT	**86.1**	**76.8**	**81.5**	**58.4**	**75.7**	–	–	–	–
	BabbleGPT	77.9	69.5	71.7	52.0	67.8				
	MLSM	69.6	65.4	74.8	52.6	65.6	–	–	–	–
	Best baseline: LTG-BERT	69.2	66.5	68.4	51.9	64.8	–	–	–	–
Strict-small	GPT-BERT	81.2	69.4	76.5	54.6	70.4	–	–	–	–
	DeBaby	74.2	63.7	73.7	54.3	66.5	–	–	–	–
	BabyLlama-2	71.8	63.4	70.2	51.5	64.2	–	–	–	–
	Best baseline: BabyLlama	69.8	59.5	63.3	50.7	61.6	–	–	–	–
Multimodal	GIT-1vd125	66.5	60.9	65.6	52.2	61.3	51.9	**57.8**	48.1	52.6
	Wake/Sleep	73.6	55.6	64.7	51.4	61.3	42.0	50.9	22.8	38.6
	Flamingo$_{\text{CL}}$	60.1	53.3	64.3	50.7	57.1	40.9	50.8	47.3	46.3
	Best baseline: Flamingo	70.9	65.0	69.5	52.7	65.2	**52.3**	51.6	**59.5**	**54.5**

Table 3: Macro averages for each benchmark across the top-performing systems (by overall score), best baseline, and skylines.

solve tasks (Dancette et al., 2021), or the information provided by different modalities may not be aggregated properly (Gadzicki et al., 2020). Furthermore, even if there are synergistic effects from multimodal or paired inputs, such as gains in learning sample efficiency, these gains can be ephemeral given more training time (Zhuang et al., 2024).

While this year's *Multimodal* track presents what is essentially a negative result, we hope that our multimodal resources lower the barrier to entry for future research in this area. Effective methods in this space remain an unsolved challenge.

5.3 Outstanding Paper Awards

We presented Outstanding Paper awards to "From Babble to Words: Pre-Training Language Models on Continuous Streams of Phonemes" (Goriely et al., 2024) and "Exploring the effect of variation sets on language model training efficiency" (Haga et al., 2024).

We selected Goriely et al. (2024) for its exploration of phonology, the study of sound or sign patterns in language, to inform tokenization. The authors incorporated phonemes into tokenization by converting raw text into phonemic transcriptions using the `phonemizer` package (Bernard and Titeux, 2021). They carefully ablate character-based, whitespace, and phoneme-aware tokenization schemes, ultimately arriving at a negative result: the standard BPE tokenization algorithm (Sennrich et al., 2016) outperforms other tokenization schemes on BabyLM's text benchmarks. However, as one might expect, phoneme-aware tokenization allows models to perform better at tasks that require phonological knowledge, such as the recognition of plausible pseudowords, or transcriptions of words that are slightly mispronounced.

Haga et al. (2024) tackle the observation from prior work that child-directed speech improves the efficiency of training language models for certain downstream tasks, such as semantic extraction (You et al., 2021) and learning of syntactic structure (Mueller and Linzen, 2023). They hypothesize that the benefits from training on child-directed speech could be due to the existence of variation sets—consecutive rephrasings of the same sentence—which are common in child-directed speech. They construct synthetic variation sets by prompting GPT-4 for paraphrases of sentences selected from CHILDES. Haga et al. find that changing the proportion of synthetic variation sets in the training data can indeed improve the performance of language models on BabyLM's evaluation tasks, although the exact characterization of this relationship remains unclear. We selected Haga et al. (2024) for the novel connections it makes

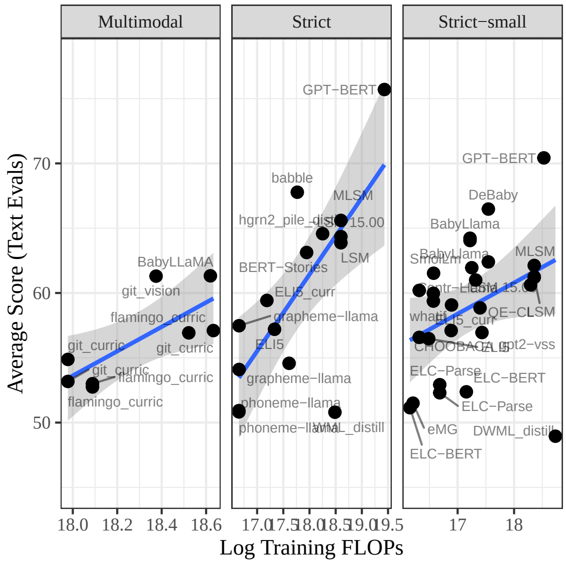


Figure 3: The relationship between training FLOPs and final score.

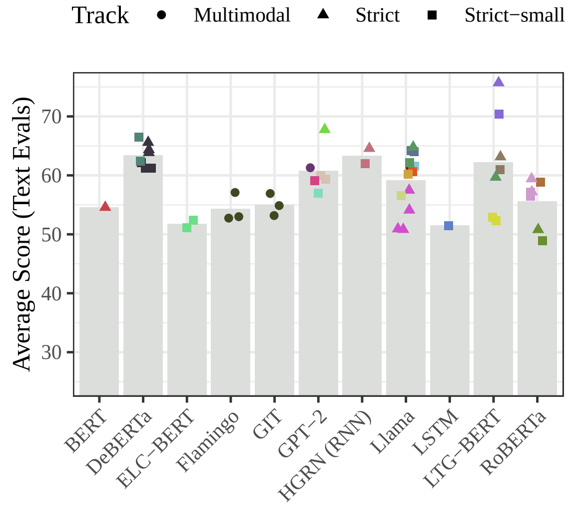


Figure 4: Scores aggregated by backbone architecture. Colors indicate different submissions.

between language modeling and specific theories from cognitive science.

6 Discussion

In this section, we discuss several trends in this year's submissions (§6.1–6.3) and spotlight approaches (§6.4) which we believe point the way towards novel and interesting work in this area.

6.1 Compute Budget

Although we did not collect systematic metadata about last year's models, we observed that our top-performing submissions tended to be more resource-intensive, particularly in the sense that winning models were trained on a large number of epochs. This raised questions about whether their high performance was due to architectural innovations or a large compute budget. We investigate this issue further in Figure 3, by visualizing the relationship between models' performance on our text-only evaluations, and their total training FLOPs. We observe a positive relationship across all three tracks. To test this relationship, statistically, we fit a linear mixed-effects regression model using the `lmer4` package in `R`, with the average score on the text evaluations as our response variable, and log training FLOPs, backbone architecture and track as covariates. We included random slopes corresponding with the model's submission ID number, which indicates the research group that submitted it. We did not include interactions between the fixed effects or random slopes due to convergence issues with the model. Inspecting the fitted model, we find that more training FLOPs leads to better performance ($\beta = 2.7, p < 0.01$), as expected.

6.2 Backbone Architecture

In Figure 4, we visualize the averaged text evaluation score broken down by each submission's backbone architecture. Relative to last year, we received more submissions using Llama. DeBERTa and HGRN (a type of RNN) lead to the highest average scores, while the highest-scoring individual models were all based on LTG-BERT, similar to last year. To test the impact of the backbone model, we inspected the fixed effects associated with model architecture from the linear regression model described above. We found that no level of backbone architecture leads to statistically significant effects for $\alpha = 0.05$, however, we did find large coefficients and smaller p values for several model architectures including DeBERTa ($\beta = 9.1, p = 0.06$), GPT-2 ($\beta = 8.5, p = 0.07$), Llama ($\beta = 7.7, p = 0.07$), and LTG-BERT ($\beta = 8.5, p = 0.06$).

Our interpretation of this result is that there are likely benefits from certain backbone architectures, but that these effects might not be strong enough to be picked up in a statistical analysis of 64 models. Interestingly, recent work has noted that different architectures and training setups often tend to converge to neural representations with similar properties and capabilities (Huh et al., 2024), and we

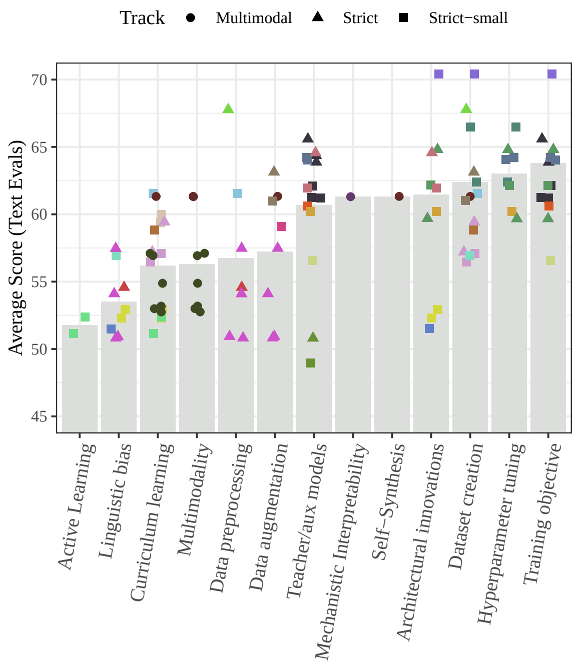


Figure 5: Scores on the BabyLM challenge, aggregated by approach. Colors indicate different submissions, which are plotted twice if they use more than one approach. Axes are zoomed to show variation in the 45-60 range more clearly.

speculate that a similar property might hold for the best models in this year's competition.

Furthermore, different backbone architectures clearly have different variances in average text evaluation score (see Figure 4). This exposes another axis of architecture quality: robustness in training. For example, in this year's competition, DeBERTa (He et al., 2021) had high average scores, compared to other architectures, and low variance between scores in submissions. The winning architecture this year was based on LTG-BERT, but LTG-BERT also had the highest variance among all backbone architectures. This suggests that picking the "best" architecture might involve trading off between architectures that can achieve high scores and architectures that are straightforward to optimize and result in lower variance.

6.3 Common Methods

In Figure 5 we visualize the models based on the approaches they employed. Each participant selected the categories that best fit their model, and categories were largely based on the typology of approaches we designed for analyzing the results of last year's challenge, however, we also let participants write-in approaches that we did not list.[9] Note that models are counted twice if they use more than one approach.

We find that modifications with the training objective, dataset creation, hyperparameter tuning, and architectural innovations lead to the highest average scores, although the latter also leads to a lot of variance across models. As with last year, curriculum learning, while popular, did not lead to high scores, on average. To investigate these trends more rigorously, we fit a mixed effects linear regression model in `lme4`. Our response variable was the average score for text-based evaluations, our covariates were dummy-coded variables indicating the approaches used for each model. We also included random intercepts associated with each submission ID number, corresponding to the research group that created the model. We did not include the interactions between the dummy variables due to convergence issues with the model. We found effects to be significant at $\alpha = 0.05$ for four approaches: training objective innovations ($\beta = 4.5, p < 0.001$), dataset creation ($\beta = 4.8, p < 0.05$), architectural innovations ($\beta = 3.5, p < 0.05$), and linguistic bias ($\beta = -7.3, p < 0.001$). Note that all coefficients are positive except for linguistic bias, meaning that this approach lead to *lower* scores. We also found a negative effect for curriculum learning ($\beta = -3.6, p = 0.055$), although the effect is not significant at the $\alpha = 0.05$ level. That being said, Figure 5 suggests that curriculum learning is not an effective strategy for improving language models, at least in the BabyLM setting.

6.4 Spotlighted Approaches

In this section, we highlight trends and new approaches used in this year's submissions.

Recurrent Neural Networks (RNNs) RNNs (Elman, 1990) made their debut in the BabyLM competition this year. The most effective RNN approach used the HGRN architecture (Qin et al., 2023), an RNN that adds complex forget gates on top of the Gated Recurrent Unit (GRU) architecture (Cho et al., 2014). As we noted in §6.2, the backbone architecture, including both RNNs and Transformers, did not have a statistically signifi-

[9] Although some participants wrote "controlled experiments" and "evaluation methods," we removed these from our visualization, as every team that submitted a model technically used these approaches.

cant impact on the models' performance on downstream evaluations, which is to say that the average performances across the best architectures were close. Nevertheless, RNNs and Transformers do have many differences, including their ability to express complex functions and the cost of performing inferences (Merrill et al., 2020; Merrill and Sabharwal, 2024). Because RNNs may be better equipped to model human language at an algorithmic level and may be more compute effective in certain settings, it was a notable finding from this year's challenge that their performance is roughly equivalent to that of many Transformers.

Synthetic Data Several contestants explored using LLMs to create synthetic training data with simple vocabularies and sentences. For example, Haga et al. (2024), used GPT-4 to create variation sets—synthetic data that was inspired by rephrases in child-directed speech. Theodoropoulos et al. (2024) extended the TinyStories approach (Eldan and Li, 2023), sampling a dataset of stories using the vocabulary of a three to four-year-old child by prompting GPT-4.

Corpus Construction Since we allowed contestants to construct their own datasets, many submissions made adjustments to the baseline BabyLM corpus. Common approaches included adding data with simpler sentences and shorter words (Ghanizadeh and Dousti, 2024) or data better suited to certain downstream evaluations (Charpentier and Samuel, 2024). Edman et al. (2024) viewed training corpus construction from the perspective of second language learning, skewing the training data towards sources that explain the rules of a language.

Auxiliary Models Explorations of auxiliary models and knowledge distillation were largely based on the BabyLlama approach introduced in last year's BabyLM challenge (Tastet and Timiryasov, 2024; Yam and Paek, 2024). BabyLlama (Timiryasov and Tastet, 2023b) trains an ensemble of causal language models on a dataset and then distills the ensemble into one final model via knowledge distillation (Hinton et al., 2015). Experiments revealed that BabyLlama's two-step training approach definitively outperforms simply training one causal language model (Tastet and Timiryasov, 2024). Berend (2024) used an extra training phase before pretraining, where the model learned to recover the sparsely encoded latent representation of an auxiliary model.

Tokenization Along with RNNs, a new trend this year was linguistically inspired tokenization (Goriely et al., 2024; Bunzeck et al., 2024). Teams explored how graphemes and phonemes could be incorporated into the language model tokenization pipeline. The primary benefit of adding graphemes and phonemes is to allow language models to perform tasks related to morphology or phonology (how words look and sound): areas where language models previously were limited (Lavechin et al., 2023). Grapheme and phoneme-aware tokenization schemes did not seem to help language models on the base BabyLM evaluation tasks.

Multi-objective training A highly successful approach across several submissions was using multiple objectives during training. GPT-BERT and AntLM, discussed in §5.2, used different methods to combine the masked and causal language modeling objectives, and both were highly successful compared to other submissions.

Training Objective Curricula Finally, a promising variant of curriculum learning this year involved creating curricula over training objectives. Salhan et al. (2024) selectively masked different parts of speech for masked language modeling over the course of training. This approach goes beyond changing the data order, which was the approach used in most curriculum learning submissions we received. We encourage participants for next year's challenge interested in curriculum learning to think beyond data order.

7 Conclusion

The second BabyLM Challenge has demonstrated that significant progress can be made in data-efficient language modeling through community-driven research efforts. With 31 submissions from 17 countries, the challenge revealed several key insights: innovations in model architecture, training objectives, and dataset construction proved particularly effective, with GPT-BERT, a hybrid causal-masked language model architecture, emerging as the strongest approach for the *Strict* and *Strict-Small* tracks. However, the strong correlation between training FLOPs and performance suggests that computational resources remain a crucial factor even in low-data settings.

While this year's challenge added a multimodal track, in an attempt to model grounded language learning environments, no submissions outperformed the baselines in this track. This suggests that effectively integrating visual information during language learning remains a significant challenge for current architectures. This year's challenge also featured emerging research directions not present in the previous iteration, with participants exploring linguistically-motivated tokenization strategies and revisiting recurrent neural architectures.

Looking ahead, we envision the BabyLM Challenge continuing to evolve and expand its scope beyond text-only and vision-language tracks. We hope that future iterations will explore additional modalities. such as speech, and extend to more languages, better reflecting the important fact that human language development proceeds equally well in any natural language. By broadening the challenge's focus while maintaining its core emphasis on data efficiency, we aim to inspire novel approaches that bridge the gap between artificial and human language learning. The strong participation and innovative solutions seen in this year's challenge suggest that the BabyLM community is well-positioned to tackle these ambitious goals, ultimately working toward language models that better reflect the efficiency and adaptability of human language acquisition.

Acknowledgments

We would like to thank the organizers of CoNLL 2024 for providing us with a venue to present BabyLM. We would also like to thank the participants of the BabyLM Challenge for their innovative submissions, engagement, and contributions to the evaluation pipeline and reviewing process.

Author Contributions

- **Primary organizers:** Alex Warstadt, Ethan Wilcox, Leshem Choshen, Aaron Mueller, Chengxu Zhuang, Michael Hu, Candace Ross
- **Pipeline implementation and maintenance:** Aaron Mueller
- **Baseline model training**: Chengxu Zhuang, Aaron Mueller
- **Publicity and communications with participants:** Leshem Choshen, Ethan Wilcox, Aaron Mueller, Michael Hu, Candace Ross
- **Training dataset compilation:** Alex Warstadt, Candace Ross, Chengxu Zhuang
- **Guidance on concept and workshop organization:** Ryan Cotterell, Tal Linzen, Adina Williams
- **Reviewing submissions:** Alex Warstadt, Ethan Wilcox, Leshem Choshen, Aaron Mueller, Michael Hu, Candace Ross
- **Initial draft of findings paper:** Alex Warstadt, Ethan Wilcox, Leshem Choshen, Aaron Mueller, Michael Hu, Candace Ross
- **Editing:** All authors

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Giusepppe Attardi. 2015. Wikiextractor. https://github.com/attardi/wikiextractor.

Eden Bensaid, Mauro Martino, Benjamin Hoover, Jacob Andreas, and Hendrik Strobelt. 2021. Fairytailor: A multimodal generative framework for storytelling. *CoRR*, abs/2108.04324.

Gábor Berend. 2024. Integrating quasi-symbolic conceptual knowledge into language model pre-training. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

Elika Bergelson and Daniel Swingley. 2013. The acquisition of abstract words by young infants. *Cognition*, 127(3):391–397.

Mathieu Bernard and Hadrien Titeux. 2021. Phonemizer: Text to phones transcription for multiple languages in python. *Journal of Open Source Software*, 6(68):3958.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2024. Graphemes vs. phonemes: Battling it out in character-based language models. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

Erin Campbell, Robyn Casillas, and Elika Bergelson. 2024. The role of vision in the acquisition of words: Vocabulary development in blind toddlers. *Developmental Science*, 27(4):e13475.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts bert. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. Bert or gpt: Why not both? In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259.

BNC Consortium. 2007. *The British National Corpus, XML Edition*. Oxford Text Archive.

Corentin Dancette, Rémi Cadène, Damien Teney, and Matthieu Cord. 2021. Beyond question-based biases: Assessing multimodal shortcut learning in visual question answering. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1554–1563.

Lukas Edman, Lisa Bylinina, Faeze Ghorbanpour, and Alexander Fraser. 2024. Are babylms second language learners? In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

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

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211. Wiley Online Library.

Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetzsche. 2020. Early vs late fusion in multimodal convolutional neural networks. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–6.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Dedre Gentner. 1982. Why nouns are learned before verbs: Linguistic relativity versus natural partitioning. *BBN report; no. 4854*.

Martin Gerlach and Francesc Font-Clos. 2018. A standardized Project Gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Computing Research Repository*, arXiv:1812.08092.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies*, 22(1). Number: 126 tex.pubmedid: 33285901.

Mohammad Amin Ghanizadeh and Mohammad Javad Dousti. 2024. Towards data-efficient language models: A child-inspired approach to language learning. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

Jill Gilkerson, Jeffrey A. Richards, Steven F. Warren, Judith K. Montgomery, Charles R. Greenwood, D. Kimbrough Oller, John HL Hansen, and Terrance D. Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

J.J. Godfrey, E.C. Holliman, and J. McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520 vol.1.

Zebulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. From babble to words: Pre-training language models on continuous streams of phonemes. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. 2023. Textbooks are all you need.

Akari Haga, Akiyo Fukatsu, Miyu Oba, Arianna Bisazza, and Yohei Oseki. 2024. Babylm challenge: Exploring the effect of variation sets on language model training efficiency. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks principle: Reading children's books with explicit memory representations. In *4th International Conference on Learning Representations*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

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

Philip A. Huebner and Jon A. Willits. 2021. Using lexical context to discover the noun category: Younger children have it easier. In Kara D. Federmeier and Lili Sahakyan, editors, *The Context of Cognition: Emerging Perspectives*, volume 75 of *Psychology of learning and motivation*, pages 279–331. Academic Press. ISSN: 0079-7421.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The platonic representation hypothesis.

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*.

Carina Kauf and Anna Ivanova. 2023. A better way to do masked language model scoring. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 925–935, Toronto, Canada. Association for Computational Linguistics.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. 2020. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981.

Marvin Lavechin, Yaya Sy, Hadrien Titeux, María Andrea Cruz Blandón, Okko Räsänen, Hervé Bredin, Emmanuel Dupoux, and Alejandrina Cristia. 2023. Babyslm: language-acquisition-friendly benchmark of self-supervised spoken language models. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, Interspeech, pages 4588–4592. International Speech Communication Association. Publisher Copyright: © 2023 International Speech Communication Association. All rights reserved.; Annual Conference of the International Speech Communication Association, INTERSPEECH ; Conference date: 20-08-2023 Through 24-08-2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Pierre Lison and Jörg Tiedemann. 2016a. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Pierre Lison and Jörg Tiedemann. 2016b. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Colleen McDonough, Lulu Song, Kathy Hirsh-Pasek, Roberta Michnick Golinkoff, and Robert Lannon. 2011. An image is worth a thousand words: Why nouns tend to dominate verbs in early word learning. *Developmental science*, 14(2):181–189.

William Merrill and Ashish Sabharwal. 2024. The expressive power of transformers with chain of thought. In *The Twelfth International Conference on Learning Representations*.

William Merrill, Gail Weiss, Yoav Goldberg, Roy Schwartz, Noah A. Smith, and Eran Yahav. 2020. A formal hierarchy of RNN architectures. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 443–459, Online. Association for Computational Linguistics.

Kanishka Misra. 2022. minicons: Enabling flexible behavioral and representational analyses of transformer language models. *CoRR*, abs/2203.13112.

Aaron Mueller and Tal Linzen. 2023. How to plant trees in language models: Data and architectural effects on the emergence of syntactic inductive biases. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11237–11252, Toronto, Canada. Association for Computational Linguistics.

Nikita Nangia and Samuel R. Bowman. 2019. Human vs. muppet: A conservative estimate of human performance on the GLUE benchmark. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4566–4575, Florence, Italy. Association for Computational Linguistics.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. 2023.

RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14048–14077, Singapore. Association for Computational Linguistics.

Miguel Pérez-Pereira and Josefa Castro. 1992. Pragmatic functions of blind and sighted children's language: A twin case study. *First Language*, 12(34):17–37.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020a. Connecting vision and language with localized narratives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 647–664. Springer.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020b. Connecting vision and language with localized narratives. In *ECCV*.

Zhen Qin, Songlin Yang, and Yiran Zhong. 2023. Hierarchically gated recurrent neural network for sequence modeling. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Suchir Salhan, Richard Diehl Martinez, Zebulon Goriely, and Paula Buttery. 2024. Less is more: Pre-training cross-lingual small-scale language models with cognitively-plausible curriculum learning strategies. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018a. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018b. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Alvin Wei Ming Tan, Sunny Yu, Bria Long, Wanjing Anya Ma, Tonya Murray, Rebecca D. Silverman, Jason D. Yeatman, and Michael C. Frank. 2024. Devbench: A multimodal developmental benchmark for language learning.

Jean-Loup Tastet and Inar Timiryasov. 2024. Babyllama-2: Ensemble-distilled models consistently outperform teachers with limited data. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

Nikitas Theodoropoulos, Giorgos Filandrianos, Vassilis Lyberatos, Maria Lymperaiou, and Giorgos Stamou. 2024. Berttime stories: Investigating the role of synthetic story data in language pre-training. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5238–5248.

Inar Timiryasov and Jean-Loup Tastet. 2023a. Baby Llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Inar Timiryasov and Jean-Loup Tastet. 2023b. Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. 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. In *Advances in Neural Information Processing Systems*, volume 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP:*

Analyzing and Interpreting Neural Networks for NLP, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*.

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.

Ethan Gotlieb Wilcox, Michael Hu, Aaron Mueller, Tal Linzen, Alex Warstadt, Leshem Choshen, Chengxu Zhuang, Ryan Cotterell, and Adina Williams. 2024. Bigger is not always better: The importance of human-scale language modeling for psycholinguistics.

Hong Meng Yam and Nathan J Paek. 2024. Teaching tiny minds: Exploring methods to enhance knowledge distillation for small language models. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

Guanghao You, Balthasar Bickel, Moritz M Daum, and Sabine Stoll. 2021. Child-directed speech is optimized for syntax-free semantic inference. *Scientific Reports*, 11(1):16527.

Xinru Yu, Bin Guo, Shiwei Luo, Jie Wang, Tao Ji, and Yuanbin Wu. 2024. Antlm: Bridging causal and masked language models. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics.

Chengxu Zhuang, Evelina Fedorenko, and Jacob Andreas. 2024. Visual grounding helps learn word meanings in low-data regimes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1311–1329, Mexico City, Mexico. Association for Computational Linguistics.

A Text Only Datasets

CHILDES. The Child Language Data Exchange System (CHILDES; MacWhinney, 2000) is a multilingual database compiling transcriptions from numerous researchers of adult–child interactions in a range of environments, from structured laboratory activities to the home. Huebner and Willits (2021) further process CHILDES, selecting only interactions with American English-speaking children ages 0–6, removing all child utterances, and tokenizing the data. The resulting dataset[10] contains about 5M words.

British National Corpus. The BNC (Consortium, 2007) is a 100M word multi-domain corpus of British English from the second half of the 20th century. We select only the dialogue portion of the corpus, totaling about 10M words.

Children's Book Test. CBT is a compilation of over a hundred children's books from Project Gutenberg by Hill et al. (2016). The dataset was originally released with a set of questions for testing named entity prediction, which we do not include in the pretraining data.

Children's Stories Text Corpus. This dataset consists of manually selected children's stories from Project Gutenberg. It was compiled by Bensaid et al. (2021) for the development of a story generation system.

Project Gutenberg. The Standardized Project Gutenberg Corpus (Gerlach and Font-Clos, 2020) is a curated and preprocessed selection of over 50k literary books in the public domain from Project Gutenberg totaling over 3B tokens.[11] This distribution comes with extensive metadata that allows us to filter texts by language and date.

OpenSubtitles. This dataset (Lison and Tiedemann, 2016b) is a compilation of publicly available subtitles from TV and movies on a third-party website.[12] We use only the English portion.

Wikipedia. Wikipedia is a volunteer-authored encyclopedia hosted by the Wikimedia Foundation. We use only the English portion.

Simple English Wikipedia. Simple English is classified as a separate language in Wikipedia, thus the texts here are disjoint from those in English Wikipedia. The texts use shorter sentences and high-frequency vocabulary and avoid idioms.

Switchboard Corpus. The Switchboard Corpus (Godfrey et al., 1992) is a collection of transcribed telephone conversations between pairs of strangers. We accessed the text through the Switchboard Dialog Act Corpus (Stolcke et al., 2000).

A.1 Text–Image Datasets

The corpus for the *Multimodal* track consisted of 50M words from the above datasets, as well as 50M more from image-caption datasets. These include the following:

Localized Narratives. Localized Narratives (Pont-Tuset et al., 2020a) is an image-caption dataset. Images are labeled by human annotators; the annotators were asked to describe an image with their voice while hovering their mouse over the region being described. We use the MS-COCO and Open Images subsets.

Conceptual Captions. Conceptual Captions (Sharma et al., 2018b) is an image-capture dataset consisting of automatically scraped and filtered images and captions/annotations from billions of web pages.

[10] https://github.com/phueb/BabyBERTa/blob/master/data/corpora/aochildes.txt
[11] https://gutenberg.org/
[12] http://opensubtitles.org/

B Evaluation Data Details

As described in Section 4.1, we filtered out evaluation examples containing words that did not appear at least twice in both the *Strict-Small* and *Multimodal* pretraining corpora. Here, we present the number of training and test examples for each evaluation task after filtering.

Note that we only control for lexical content: other factors, such as sentence length, syntactic complexity, and overall linguistic style, remain distinct between our corpus and these tasks. In the future, it would be helpful for researchers to focus on designing tasks on which both children *and* language models can be reasonably evaluated.

Note, too, that this filtering step implies that we cannot directly compare results obtained from the BabyLM Challenge to prior evaluations using the full datasets. We also cannot directly compare to results from last year's challenge, though we believe the overlap between the evaluation sets across the BabyLM Challenges is likely high.

Task	Subtask	|Train|	|Test|
BLiMP	–	–	59875
BLiMP Supplement	Hypernym	–	842
	Question-Answer Congruence (easy)	–	64
	Question-Answer Congruence (tricky)	–	165
	Subject-Auxiliary Inversion	–	3867
	Turn-taking	–	280
(Super)GLUE	CoLA	8551	522
	SST-2	67349	436
	MRPC	3668	204
	QQP	363846	20215
	MNLI	392702	4908
	MNLI-mismatched	–	4916
	QNLI	104743	2732
	RTE	2490	139
	BoolQ	9427	1635
	MultiRC	27243	2424
	WSC	554	52
EWoK	Agent Properties	–	2210
	Material Dynamics	–	770
	Material Properties	–	170
	Physical Dynamics	–	120
	Physical Interactions	–	556
	Physical Relations	–	818
	Quantitative Properties	–	314
	Social Interactions	–	294
	Social Properties	–	328
	Social Relations	–	1548
	Spatial Relations	–	490

Task	Subtask	|Train|	|Test|
VQA	–	–	25230
Winoground	–	–	746
DevBench	Visual Vocabulary	–	433
	Test of Receptive Grammar (TROG)	–	79
	THINGS	–	12340

Table 4: Number of training and test examples for each BabyLM evaluation task. We present the number of examples for the text-only tasks (left) and the multimodal tasks (right). We show the number of examples *after* filtering based on the pre-training corpus vocabulary (Section 4.1). Note that only the (Super)GLUE has training examples; the rest of the tasks are zero-shot.

C Subtask Results

Here, we present a more detailed breakdown of results by subtask. Each task has a subsection containing a table where results are described, as well as a textual description containing and overview of the main takeaways for each task.

C.1 BLiMP and BLiMP Supplement

GPT-BERT was the best-performing model on the BLiMP tasks in both the *Strict* and *Strict-Small* tracks. The only subtask where it did not perform best among all models was for Hypernym, where the LTG-BERT baseline was best. BabbleGPT and AntLM were the runners-up in the *Strict* track, whereas DeBaby and BabyLlama-2 were the runners-up in the *Strict-Small* track. In general, submissions to the *Multimodal* track did not consistently outperform the baseline models; Wake/Sleep outperformed the best baseline (Flamingo) on BLiMP, but no submission outperformed Flamingo on the BLiMP Supplement.

In general, the average BLiMP score across subtasks was effective in distinguishing between high- and low-performing systems: there is high variance across submissions, and those that perform best on BLiMP also tend to perform comparatively well on other tasks.

Similarly to last year, we observe that the HYPERNYM test suite is beyond the ability of language models of this scale. All models (including last year's skylines) perform very close to chance, suggesting either that their preferences are virtually random guessing, or they show systematic biases that essentially cancel out due to counterbalancing in the test data. However, we hesitate to conclude that these models have no knowledge of lexical entailment relations for two reasons: First, these test sentences are somewhat unnatural logical statements that are out-of-domain for the models; and second, there is less reason *a priori* to believe that logically invalid statements have lower probabilities than valid statements.

Among the QUESTION–ANSWER CONGRUENCE test suites, we find that the "tricky" set is still highly discriminative, probably due in part to its adversarial nature. This tells us that most models are easily fooled by locally coherent distractor answers and pay too little attention to cross-sentential long-distance dependency between a *wh*-word and a congruent answer. Only the top-performing models in the *Strict* track score better than chance, and the RoBERTa skyline outperforms all models by a wide margin.

The tests for SUBJECT–AUXILIARY INVERSION are relatively easy: the best models reach near-perfect accuracy, and all models score relatively high compared to other test suites.

Finally, TURN TAKING is highly discriminative, with some models performing at or near chance, while the best model achieves accuracy over 90%.

		BLiMP	BLiMP Supplement					
	Model	Macro average	Macro average	Hypernym	Q–A congruence (easy)	Q–A congruence (tricky)	Subject–aux inversion	Turn taking
Strict	GPT-BERT	**86.1**	**76.8**	48.8	**90.6**	**59.4**	**96.3**	**88.9**
	BabbleGPT	77.8	69.5	47.9	81.2	52.1	81.9	84.3
	AntLM	74.9	66.0	49.3	79.7	43.6	78.3	79.3
	Base baseline: LTG-BERT	69.2	66.5	**55.0**	75.0	53.3	87.5	61.4
Strict-small	GPT-BERT	81.2	69.4	47.1	73.4	54.5	86.3	85.7
	DeBaby	74.2	63.7	53.3	79.7	49.1	84.1	52.1
	BabyLlama-2	73.2	63.1	49.8	59.4	41.2	90.3	75.0
	Best baseline: BabyLlama	69.8	59.5	49.6	54.7	41.2	86.0	66.1
Multimodal	Wake/Sleep	73.6	55.6	49.5	50.0	30.9	85.3	62.1
	GIT-1vd125	66.5	60.9	48.2	57.8	44.2	86.5	67.9
	$\mathrm{GIT}_{\mathrm{CL}}$	64.0	51.2	48.9	50.0	20.0	83.7	53.6
	Best baseline: Flamingo	70.9	65.0	48.8	75.0	43.6	86.2	71.4

Table 5: BLiMP Supplement accuracies for each subtask for the top performing systems (by overall score), best baseline, and skylines. For each subtask, we mark the best performing system for each track, and the **best** performing system overall.

C.2 GLUE/SuperGLUE

Scores on (Super)GLUE tasks (Table 6) show that GPT-BERT is the best-performing system in both the *Strict* and *Strict-Small* tracks. Notably, its performance in the *Strict-Small* track is better than the runners-up in the *Strict* track, suggesting that this approach is highly data-efficient and/or well-tuned for small-scale language modeling. BabbleGPT and AntLM were again the runners-up for (Super)GLUE in the *Strict* track, and DeBaby was again the runner-up for the *Strict-Small* track. MLSM is now second runner-up in the *Strict-Small* track. Once again, no submissions outperformed the best baseline (Flamingo) in the *Multimodal* track. This largely confirms findings from the BLiMP and BLiMP Supplement tasks.

	Model	**Macro average**	**CoLA**	**SST-2**	**MRPC**	**QQP**	**MNLI**	**MNLI-mm**	**QNLI**	**RTE**	**BoolQ**	**MultiRC**	**WSC**
Strict	GPT-BERT	**81.5**	**62.4**	**94.0**	**94.4**	**89.1**	**85.2**	**85.3**	**90.8**	**69.1**	**78.4**	**73.3**	**75.0**
	Babble-GPT	71.7	37.8	89.4	83.8	84.0	75.3	76.4	82.9	66.2	63.7	65.1	63.5
	AntLM	66.3	22.2	89.4	84.9	84.2	74.8	74.4	83.2	55.4	65.8	59.9	34.6
	Best baseline: LTG-BERT	68.4	34.6	91.5	83.1	86.7	77.7	78.1	78.2	46.8	61.7	52.6	61.5
Strict-small	GPT-BERT	76.5	48.9	92.2	91.5	87.1	80.2	80.5	86.4	64.0	72.5	69.3	69.2
	DeBaby	73.7	41.8	89.2	91.2	86.6	78.1	77.6	85.5	69.8	71.1	64.2	55.8
	MLSM	73.3	45.2	90.6	82.2	86.6	76.4	77.4	84.7	60.4	69.4	67.6	65.4
	Best baseline: BabyLlama	63.3	2.2	86.2	82.0	83.6	72.4	74.2	82.8	49.6	65.0	60.1	38.5
Multimodal	GIT-1vd125	65.6	30.7	89.7	81.5	83.3	72.7	72.6	78.4	51.8	64.2	54.7	42.3
	Wake/Sleep	64.7	12.2	79.8	78.4	80.5	69.4	70.6	79.8	52.5	63.1	65.8	59.6
	$\text{Flamingo}_{\text{CL}}$	64.3	31.8	88.3	82.4	81.9	70.4	71.4	69.9	46.0	66.5	56.2	42.3
	Best baseline: Flamingo	69.5	36.7	90.4	84.2	85.1	75.8	76.4	83.8	60.4	69.1	60.5	42.3

Table 6: (Super)GLUE results for each subtask for the top performing systems (by overall score), best baseline, and skylines. For each subtask, we mark the best performing system for each track, and the **best** performing system overall.

C.3 Multimodal Tasks

Model	**Macro average**	**VQA**	**Winoground**
GIT-1vd125	**54.9**	51.9	**57.8**
GIT_{CL}	49.6	44.0	55.2
Wake/Sleep	46.5	42.0	50.0
Best baseline: GIT	54.8	**54.1**	55.5

Table 7: Results for the public multimodal tasks for the top performing systems (by average score), and the best baseline. For each subtask, we mark the best performing system for each track, and the **best** performing system overall.

Model	**Macro average**	**Visual Vocabulary**	**TROG**	**THINGS**
$\text{Flamingo}_{\text{CC}}$	49.0	66.4	34.2	**46.5**
GIT_{CL}	48.2	73.1	**39.5**	32.1
GIT-1vd125	48.1	**84.9**	35.5	23.8
Best baseline: Flamingo	**59.5**	80.7	38.2	32.6

Table 8: Results for the DevBench tasks for the top performing systems (by average score), and the best baseline. For each subtask, we mark the best performing system for each track, and the **best** performing system overall.

Towards Data-Efficient Language Models: A Child-Inspired Approach to Language Learning

Mohammad Amin Ghanizadeh and Mohammad Javad Dousti
Department of Electrical and Computer Engineering,
College of Engineering, University of Tehran, Tehran, Iran
{ghanizadeh.amin,mjdousti}@ut.ac.ir

Abstract

In this work, we explain our approach employed in the BabyLM Challenge, which uses various methods of training language models (LMs) with significantly less data compared to traditional large language models (LLMs) and are inspired by how human children learn. While a human child is exposed to far less linguistic input than an LLM, they still achieve remarkable language understanding and generation abilities. To this end, we develop a model trained on a curated dataset consisting of 10 million words, primarily sourced from child-directed transcripts. The 2024 BabyLM Challenge initial dataset of 10M words is filtered to 8.5M. Next, it is supplemented with a randomly selected subset of TVR dataset consisting of 1.5M words of television dialogues. The latter dataset ensures that similar to children, the model is also exposed to language through media. Furthermore, we reduce the vocabulary size to 32,000 tokens, aligning it with the limited vocabulary of children in the early stages of language acquisition. We use curriculum learning and is able to match the baseline on certain benchmarks while surpassing the baseline on others. Additionally, incorporating common LLM training datasets, such as MADLAD-400, degrades performance. These findings underscore the importance of dataset selection, vocabulary scaling, and curriculum learning in creating more data-efficient language models that better mimic human learning processes.

1 Introduction

Language models (LMs) have revolutionized natural language processing, demonstrating remarkable capabilities in understanding and generating human-like text. However, the training of these models typically requires vast amounts of data, often billions of words, which stands in stark contrast to how human children acquire language. The BabyLM Challenge (Choshen et al., 2024) seeks to bridge this gap by exploring methods to train LMs more efficiently, using significantly less data while still achieving high performance.

Human children develop impressive language skills despite being exposed to far less linguistic input than traditional large language models (LLMs). This observation raises intriguing questions about the efficiency of human language acquisition and the potential for more data-efficient machine-learning approaches. Our research addresses these questions by mimicking the human language acquisition process.

In this work, we present our approach to the BabyLM Challenge, focusing on developing a model that can learn effectively from a dataset more closely aligned with the linguistic experiences of a young child. Our primary contributions are as follows:

1. Dataset curation: We carefully curated a dataset of 10 million words, primarily sourced from child-directed transcripts. This dataset was then refined to 8.5 million words and supplemented with 1.5 million words of television dialogue, acknowledging the role of media exposure in modern language acquisition.

2. Vocabulary scaling: To better mimic the limited vocabulary of children in the early stages of language acquisition, we reduced the model's vocabulary size to 32,000 tokens. This constraint forces the model to learn more efficient representations and generalization strategies. Also, this vocab size is similar to the tokenizer used in Llama models (Touvron et al., 2023).

3. Model architecture: We adopted the SmolLM model (Allal et al., 2024) which uses a decoder-only Transformer (Brown, 2020) model with 125 million parameters, trained for 5 epochs. This relatively compact model size allows us to explore the limits of what

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 22–27
November 15-16, 2024

can be achieved with limited data and computational resources.

4. Experimental variations: We conducted several experiments to evaluate the impact of different training strategies: a) We compared model performance with and without the inclusion of television dialogue data. b) We explored the potential benefits of curriculum learning (Bengio et al., 2009), developing a method for scoring and sorting data points based on complexity. c) We investigated the impact of incorporating high-quality monolingual datasets, such as MADLAD-400 (Kudugunta et al., 2024), on model performance.

The curriculum learning implementation involved developing custom scoring functions to assess sentence complexity based on factors like word count, average word length, unique word ratio, and punctuation usage, similar to (Nagatsuka et al., 2023). These scores were then used to sort the dataset, allowing for a structured learning progression.

By focusing on dataset selection, vocabulary scaling, and curriculum learning, we present a framework for developing more efficient language models that could have significant implications for both cognitive science and practical NLP applications.

The rest of this paper is as follows. Section 2 details our methodology. Next, Section 3 presents our experimental results. After that, Section 4 discuss the implications of our findings for future research in data-efficient language model training, and Section 6 concludes the paper.

2 Methodology

Our approach to the BabyLM Challenge involves careful data preparation followed by the implementation of a curriculum learning strategy. This section details our methods for dataset curation and the subsequent application of curriculum learning.

We started with an initial dataset of approximately 10 million words, primarily sourced from child-directed transcripts. This dataset was chosen to closely mimic the linguistic input that young children typically receive during their language acquisition process.

To enhance the quality and relevance of our training data, we implemented a rigorous filtering process as explained next:

Duplicate removal: Similar to (Rae et al., 2021), we identified and removed exact duplicate sentences from the dataset. This step helps to prevent overfitting to specific phrases and ensures a more diverse linguistic input.

Content refinement: After duplicate removal, we further refined the dataset based on relevance and quality criteria. For instance, we excluded data points where the ratio of punctuation marks to total words exceeded 0.33 and removed samples with less than 10 characters, resulting in a reduced dataset of approximately 8.5 million words.

Recognizing that modern-age children often acquire language partially through media exposure, we supplemented our refined dataset with television dialogue. We carefully selected approximately 1.5 million words of television dialogue, focusing on content appropriate for and often consumed by young children.

The TV data from the TVR dataset (Lei et al., 2020) was added to our refined 8.5 million word dataset, resulting in a final training corpus of about 10 million words. The inclusion of TV dialogue adds diversity to our dataset and better reflects the varied sources of language input in a child's environment.

Following the data preparation phase, we implemented a curriculum learning approach to optimize the training process. This method is designed to present the model with progressively more complex linguistic inputs, mimicking the natural progression of language acquisition observed in human learners.

We developed a set of scoring functions to assess the complexity of each data point in our dataset. These functions evaluate various linguistic features as briefed next.

- **Word count:** A basic measure of sentence length.
- **Average word length:** An indicator of vocabulary complexity.
- **Unique word ratio:** A measure of lexical diversity within a sentence.
- **Punctuation count:** An indirect measure of syntactic complexity.

Each data point is passed through these scoring functions, generating a set of individual scores that capture different aspects of linguistic complexity.

$$score(d) = \sum_{f \in F} w_f f(d), \quad (1)$$

where d is a data point which the score is computed for, F is the set of functions used for scoring, and w_f is the weight of each scoring function, which ranges between 0 and 1. The sum of all weights should be equal to 1. We conducted experiments with various weight configurations for each function and found that the unique word count function had a greater influence on the final outcome. As a result, we assigned it a weight of 0.4, while all other functions were assigned a weight of 0.2.

Once the complexity score is calculated for each data point, we sort the entire dataset in ascending order of these scores. The sorted dataset forms the basis of our curriculum learning approach.

Training begins with the least complex data points (lowest scores). As training progresses, more complex data points are introduced. By the end of training, the model has been exposed to the full range of linguistic complexity present in the dataset. Throughout the training process, the model's learning rate decreases. Revisiting simpler examples in later epochs with a lower learning rate helps fine-tune the model's understanding of fundamental concepts while reducing the risk of overfitting.

This gradual exposure to complexity allows the model to build a foundational understanding of simpler linguistic structures before tackling more complex ones, potentially leading to more robust and efficient learning.

3 Experiments

3.1 Experiments' Setup

To evaluate the effectiveness of our approach in the BabyLM Challenge, we conducted a series of experiments designed to test various aspects of our model[1] and training methodology. Our experimental setup was guided by the goal of creating a data-efficient language model that could perform well on benchmark tasks while using significantly less training data than traditional large language models.

We trained a decoder-only transformer model with 125 million parameters. The model was trained for 5 epochs, with the best-performing checkpoint selected based on the model's performance on the validation dataset. Our vocabulary size was set to 32,000 tokens, aligning with our strategy of mimicking the limited vocabulary of children in the early stages of language acquisition. The employed hyperparameters are summarized in Table 1.

[1] https://huggingface.co/universitytehran/SmolLM-135M-10M-word

Hyperparameter	**Value**
Architecture	SmolLM
Model size	125M
Tokenizer vocab size	32,000
Batch size	32
Learning rate	5e-5
Weight decay	0.015
Learning rate scheduler	Linear
Number of decoder layers	30
Number of attention heads	9

Table 1: Model and training parameters.

	BLIMP	**BLIMP supplement**
Without TV data	69.8	57.9
With TV data	**72.2**	**59.1**
MADLAD data	68.2	55.0

Table 2: The impact of adding 1.5M words of training data from TVR and MADLAD datasets on the performance of the model.

In the rest of this section, we present the results of these experiments, providing a detailed analysis of our findings and their implications for data-efficient language model training.

3.2 Results

In our initial investigation, we explored the impact of utilizing television data as a rich linguistic resource within a constrained data environment. As shown in Table 2, incorporating transcribed text from television shows significantly enhances the model's performance on BLIMP and BLIMP Supplement benchmarks. We selected 1.5M words from the TVR and MADLAD datasets to replace with those from the original dataset, while keeping the overall dataset size unchanged. This observation suggests that the diverse language patterns, dialogues, and narratives present in television content provide valuable linguistic information that can be effectively leveraged to improve language model capabilities.

As shown in Table 3, a key finding from our

Vocab size	BLIMP	BLIMP supplement
30,000	71.1	57.3
32,000	**72.2**	**59.1**
50,000	69.0	54.4

Table 3: The impact of tokenizer vocabulary size on the performance of the model.

experiments pertains to the optimal vocabulary size for language model training. We discovered that a vocabulary size of approximately 32,000 tokens yields the best-performing models. Interestingly, both smaller and larger vocabulary sizes resulted in diminished performance compared to this optimal range. This finding highlights the importance of carefully considering vocabulary size as a crucial hyperparameter in language model development.

To further validate this observation, we trained our own tokenizer on English language data, specifically targeting a vocabulary size of 32,000 tokens. This custom tokenizer allowed us to tailor the vocabulary to our specific dataset while maintaining the optimal size identified in our experiments. All models trained with tokenizers of various sizes were trained on the same dataset, consisting of 8.5 million samples along with an additional 1.5 million samples from TV data.

The third significant finding from our research demonstrates the efficacy of curriculum learning in boosting model performance. We implemented a curriculum learning approach by assigning scores to each data point in our dataset using the scoring functions discussed earlier in our methodology. By training the model on this scored data, we observed a notable improvement in overall performance.

This curriculum learning strategy enables the model to gradually learn from simpler to more complex examples, potentially leading to more robust and generalizable language understanding. Our results suggest that carefully designed learning curricula can play a crucial role in optimizing the training process and ultimately enhancing the capabilities of language models.

In an effort to explore alternative data sources, we conducted experiments using the MADLAD (Kudugunta et al., 2024) dataset as a substitute for our initially provided dataset. For the selection of MADLAD data, we applied the same set of filters used to curate the 8.5 million word dataset. After filtering, we sampled a total of 10 million words from the MADLAD dataset. Contrary to our expectations, we observed a decrease in performance across both the BLiMP and BLiMP supplement benchmarks. Specifically, the model trained on MADLAD (Kudugunta et al., 2024) data achieved scores of 68.2 and 55.0 on these benchmarks, respectively, which were lower than the scores obtained using our original dataset.

This unexpected outcome led us to a crucial insight regarding the nature of high-quality data in language modeling. We posit that the definition of *high-quality data* may vary significantly between low-resource and rich-resource language modeling scenarios. In low-resource environments, where data scarcity is a primary constraint, the emphasis may need to be placed on data that is particularly rich in linguistic structures and diverse in its representation of the target language. Conversely, in rich-resource scenarios, the sheer volume of data might compensate for potential variations in quality.

Table 4 compares our model against the baselines. Our model outperforms or matches the baselines across all benchmarks, except for the BLiMP Supplement. Overall, our model's performance exceeds that of the best-scoring baseline.

4 Discussion

We hypothesize that data valuation and attribution methods could offer significant advantages over current data selection techniques. While not directly implemented in our study, methods such as Influence functions (Koh and Liang, 2017) , Representer point (Yeh et al., 2018), and dynamic approaches like TracIn (Pruthi et al., 2020) and HyDRA (Chen et al., 2021), or RL-based methods for data valuation (Yoon et al., 2020), show promise as potential tools for more effective data curation. These techniques, originally designed to quantify the impact of individual data points on model performance, could potentially be adapted to filter large datasets into smaller, higher-quality subsets. Unlike traditional data selection methods such as number of characters (Raffel et al., 2020), frequency (Laurençon et al., 2022), or using a blocklist (Penedo et al., 2023) that may rely on simplistic criteria, these advanced techniques could provide a more nuanced understanding of data importance. By identifying the most influential or informative samples, they might enable researchers to create more compact yet equally effective training sets. This approach could lead to reduced com-

	BLiMP	BLiMP Supplement	EWoK	GLUE	Macro Average
BabyLlama	69.8	59.5	**50.7**	63.3	60.8
LTG-BERT	60.6	**60.8**	48.9	60.3	57.7
Ours (w/o curriculum training)	71.5	58.6	50.4	62.8	60.8
Ours (w/ curriculum training)	**72.2**	59.1	**50.7**	**63.9**	**61.5**

Table 4: Comparison between our model and baselines on BLiMP (Warstadt et al., 2020), BLiMP supplement, GLUE (Wang et al., 2018), and EWoK (Ivanova et al., 2024).

putational costs, faster training times, and potentially more robust models. Furthermore, in fields where data collection is resource-intensive, such methods might guide more targeted and efficient data gathering strategies. While further research is needed to validate this hypothesis, exploring the application of these methods in data curation could open new avenues for improving the efficiency and effectiveness of machine learning pipelines.

5 Limitations

Despite the promising results, this study has several limitations. First, our approach relies on the weights used for scoring data during curriculum learning. With a different set of weights, performance may even decline compared to not using curriculum learning. Furthermore, these weights may vary across different datasets, and finding their near-optimal values could be computationally expensive. Second, the appropriate amount of TV data was selected experimentally and may differ for other datasets. Lastly, the effect of training with this procedure on downstream tasks is unclear and may negatively impact model performance in those tasks. Future research should aim to address these limitations by developing a reliable and robust method for determining score weights, selecting the appropriate portion of TV data, and assessing the influence of this approach on downstream task performance.

6 Conclusion

This study, conducted as part of the BabyLM Challenge, has yielded several significant insights into the development of data-efficient language models that more closely mimic human language acquisition. Our approach, focusing on careful dataset curation, vocabulary scaling, and curriculum learning, has demonstrated promising results in training a language model with substantially less data than traditional large language models.

These results have important implications for both cognitive science and practical NLP applications. By demonstrating that effective language models can be trained on significantly smaller datasets, our work contributes to the ongoing discussion about data efficiency in AI and machine learning. Furthermore, our findings suggest potential avenues for developing more cognitively plausible models of language acquisition, which could inform both AI research and our understanding of human language learning.

Acknowledgement

This research was supported in part by the Iran Cognitive Sciences and Technologies Council. We acknowledge the use of OpenAI ChatGPT in the writing and editing process of this manuscript.

References

Loubna Ben Allal, Anton Lozhkov, and Elie Bakouch. 2024. SmolLM - blazingly fast and remarkably powerful. https://huggingface.co/blog/smollm. Accessed: 2024-10-20.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Yuanyuan Chen, Boyang Li, Han Yu, Pengcheng Wu, and Chunyan Miao. 2021. Hydra: Hypergradient data relevance analysis for interpreting deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7081–7089.

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

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*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2024. MADLAD-400: A multilingual and document-level large audited dataset. *Advances in Neural Information Processing Systems*, 36.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020. TVR: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2023. Length-based curriculum learning for efficient pre-training of language models. *New Generation Computing*, 41(1):109–134.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33:19920–19930.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

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*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

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.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. 2018. Representer point selection for explaining deep neural networks. *Advances in neural information processing systems*, 31.

Jinsung Yoon, Sercan Arik, and Tomas Pfister. 2020. Data valuation using reinforcement learning. In *International Conference on Machine Learning*, pages 10842–10851. PMLR.

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).

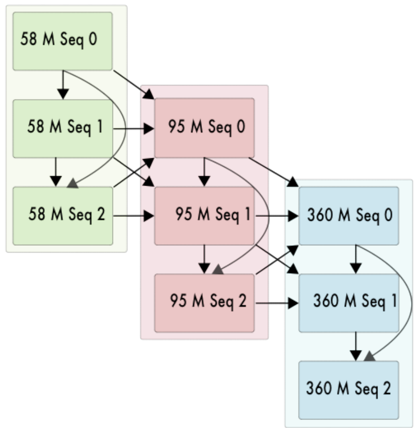


Figure 1: We train an expanding series of models using a moving window ensemble containing the previously trained models (left to right) as teachers. The model with sequence number 1 is trained on two predecessor models of smaller size and one of the same size. While models with sequence numbers 0 and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

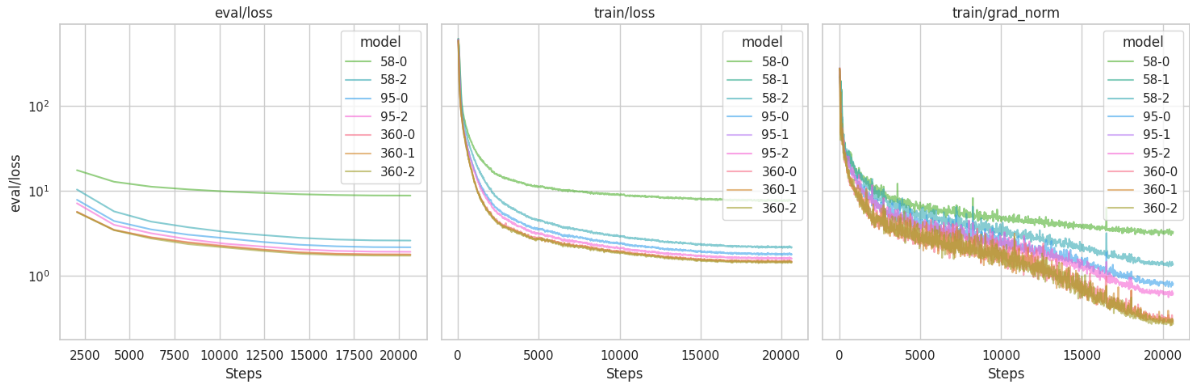


Figure 2: Evaluation and training loss along with gradient norms for models in the sequence. We note that models later in the teaching sequence and larger models have steeper decline losses than models earlier in the sequence.

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.

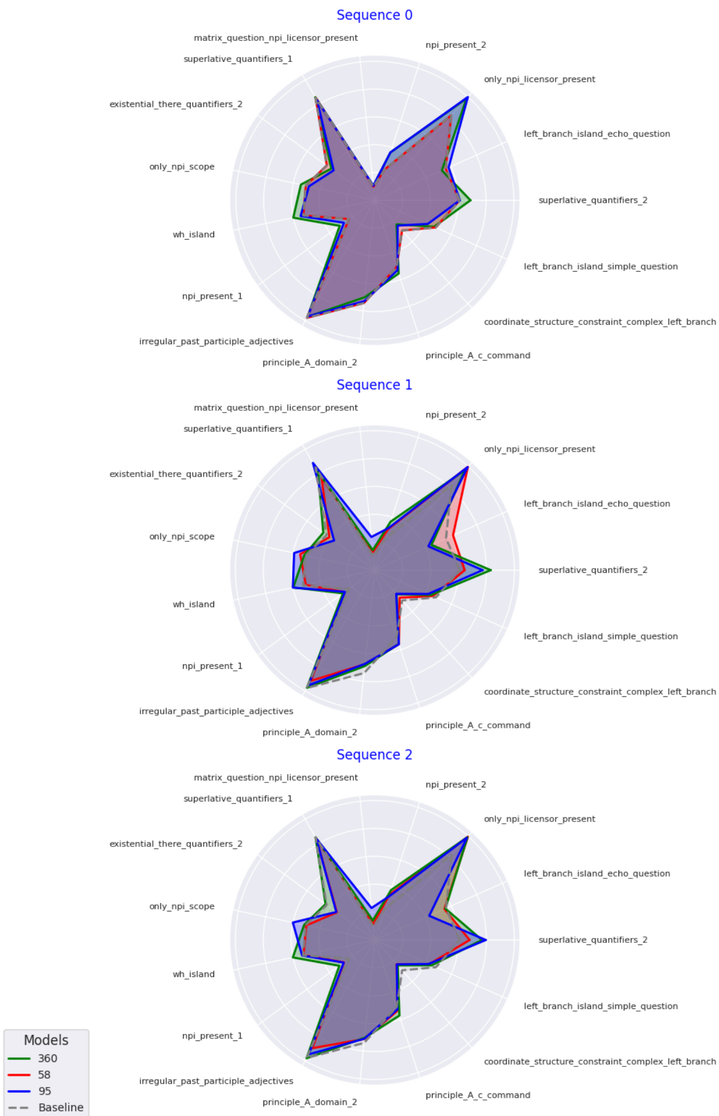


Figure 5: Blimp results for 9 models, grouped by sequence. All models were fine-tuned with standard parameters given by BabyLLM organizers except the number of epochs parameter, which was set to 3. We show the sub-tasks which have the highest variance across the models.

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.

From Babble to Words :
Pre-Training Language Models on Continuous Streams of Phonemes

Zébulon Goriely **Richard Diehl Martinez**
Andrew Caines **Lisa Beinborn** **Paula Buttery**
Department of Computer Science & Technology, University of Cambridge, U.K.
ALTA Institute, University of Cambridge, U.K.
University of Göttingen, Germany
firstname.secondname@cl.cam.ac.uk lisa.beinborn@uni-goettingen.de

Abstract

Language models are typically trained on large corpora of text in their default orthographic form. However, this is not the only option; representing data as streams of phonemes can offer unique advantages, from deeper insights into phonological language acquisition to improved performance on sound-based tasks. The challenge lies in evaluating the impact of phoneme-based training, as most benchmarks are also orthographic. To address this, we develop a pipeline to convert text datasets into a continuous stream of phonemes. We apply this pipeline to the 100-million-word pre-training dataset from the BabyLM challenge, as well as to standard language and grammatical benchmarks, enabling us to pre-train and evaluate a model using phonemic input representations. Our results show that while phoneme-based training slightly reduces performance on traditional language understanding tasks, it offers valuable analytical and practical benefits.

phonemetransformers/FromBabbleToWords (CC BY 4.0)
codebyzeb/PhonemeTransformers (CC BY 4.0)
codebyzeb/CorpusPhonemizer (CC BY 4.0)

1 Introduction

The use of orthographic text to train neural networks is so commonplace that it is considered the default. This has not always been the case.

When neural networks were first applied to language, models were primarily trained on continuous streams of phonemes or graphemes, rather than orthographic text with its written artefacts. These early neural models demonstrated a striking ability to acquire phonology, syntax and semantics (Elman, 1990; Seidenberg and McClelland, 1989; Prince and Smolensky, 1997). As technology scaled, subword representations became the dominant representation, offering key advantages such as reducing computation costs and better capturing out-of-vocabulary items (Sennrich et al., 2016). Written text became favored over speech transcriptions due to matching the domain of downstream tasks and due to the abundance of diverse texts available through web-scraping (Bansal et al., 2022). Today, "large language models" (LLMs) all use subword-based text inputs and perform impressively on a variety of language understanding tasks (Zellers et al., 2019; Hendrycks et al., 2020; Suzgun et al., 2023).

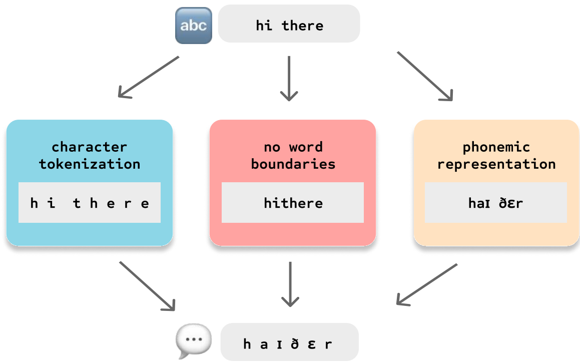


Figure 1: An illustration of all three adjustments that we make to convert text input to continuous streams of phonemes.

The success of these models on downstream tasks has motivated researchers to examine the internal representations of LLMs and analyze their ability to learn grammatical generalizations (Hewitt and Manning, 2019; Hu et al., 2020; Manning et al., 2020). However, their phonological capabilities remain understudied due to the orthographic nature of training data.

An alternative input representation for text-based language models is to use phonemes rather than graphemes, corresponding to how words are pronounced, rather than how they are written. The use of phonemes, such as those described by the International Phonetic Alphabet (IPA), as an underlying input representation, presents the following analytical and practical benefits over an orthographic representation that is the modern-day default.

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 37–53
November 15-16, 2024 ©2024 Association for Computational Linguistics

Analytical: A phoneme-based representation is useful when using language models to study the distributional properties of phonemes (Mayer, 2020) and phonological systems of languages more broadly (Eden, 2018). Many language acquisition studies prefer using phonemes as a representation that more closely represents the human learning environment, which facilitates statistical learning experiments ranging from word segmentation (Çöltekin, 2017), to past-tense formation (Kirov and Cotterell, 2018), and broader lexico-syntactic knowledge (Lavechin et al., 2023).

Practical: IPA-encoded text has been found to be beneficial for a variety of NLP tasks including lyric generation (Ding et al., 2024), text-to-speech (Sundararaman et al., 2021; Li et al., 2023) and low-resource language modeling (Leong and Whitenack, 2022). Phonemes also benefit multi-lingual language modeling by establishing a universal representation shared between languages (Feng et al., 2023; Zhu et al., 2024).

Despite the analytical and practical advantages of training language models with phonemes, a key question remains: *Can modern language model architectures encode grammatical knowledge and succeed at language understanding tasks when trained with phoneme-based representations?*

Answering this question is challenging for two reasons. First, training and evaluation data need to be provided to a model in both a phonemic and graphemic representation. Second, it is non-trivial to select the transformations to convert orthographic text into phonemic representations and to evaluate how these individually affect a model's performance across a wide variety of benchmarks.

In this work, we address these challenges as follows. We first present a method for converting training data and evaluation benchmarks into a unified IPA representation. This enables language models to be trained and evaluated on graphemic and phonemic representations of the same data. We then identify three key transformations which enable us to map from the written representation typically used to train language models to the phonemic representation often used in analytical studies (see fig. 1). Finally, we conduct a careful ablation of the three transformations: we train a language model on the same corpus of 100 million words with all combinations of the three transformations (2^3 configurations), evaluating the model's grammatical capabilities and its resulting performance on downstream language understanding tasks.

We find that large language models are powerful statistical learners capable of learning grammar from a phonemic input representation. Although we observe a decrease in performance on some tasks, the degradation is not as substantial as has been anecdotally suggested by previous studies. Our ablation studies indicate that the impact of each transformation that we use to convert orthographic text to continuous phoneme streams depends on the downstream task; tasks in the BLiMP Supplement set are particularly sensitive to the use of phonemes, while those in GLUE are sensitive to character tokenization. A deeper analysis into these ablations reveals that many evaluation instances rely on information only present in written text (such as punctuation). Finally, we take advantage of the fact that we train models using phonemic streams and evaluate our models for phonological knowledge using the BabySLM benchmark. Our models achieve the best scores on this benchmark to date.

2 Related Work

The standard input representation for training large language models consists of written text split into subword units. By contrast, studies that train models using a phonemic input representation tend to split words into individual phonemes, without word boundaries (as spoken utterances are produced continuously, without clear pauses between words).

We identify three key transformations that bring us from the standard input representation used by language models to this alternative **phoneme stream** representation:

- Character tokenization Treating each phoneme or grapheme as a token, rather than using subwords.
- Word boundary removal Removing whitespace or other word boundary cues from the input.
- Phonemic transcription Converting words to a phonemic representation.

Each transformation can be made independently or in combination, as illustrated in fig. 1.

Previous literature has extensively explored these three transformations but they have typically been studied independently and been used for different downstream purposes.

2.1 Training with Phonemes

Several language models have been trained with phonemic input (Sundararaman et al., 2021; Gale et al., 2023) but it remains a challenge to do so due to the lack of large phonemic corpora. While a number of well-known speech-based datasets include phonemic transcriptions, such as Switchboard (Godfrey et al., 1992) and TIMIT (Garofolo et al., 1993), these datasets are small compared to the trillions of tokens contained in standard language model pre-training corpora (Elazar et al., 2024). The majority of works that use phonemic representations typically rely on grapheme to phoneme conversion tools (Bisani and Ney, 2008; Hasegawa-Johnson et al., 2020) to generate coarse phonemic transliterations of text data.

It is also a challenge to evaluate the broad capabilities of language models trained with phonemes, as most benchmarks assume a graphemic representation, even some that assess phonological knowledge (Suvarna et al., 2024). One benchmark that assesses both the syntactic and phonological capabilities of language models is BabySLM (Lavechin et al., 2023). We discuss this benchmark further in section 5.1.

2.2 Character-based Language Models

The use of characters as the input representation, rather than words or subwords, has been extensively explored. Character-level language models offer a simplified input stream compared to the standard approach of training on learned subword tokens. Many studies have developed specialized architectures to train language models on characters (Jozefowicz et al., 2016; Kim et al., 2016; Ma et al., 2020; Al-Rfou et al., 2019) while other approaches seek to establish 'token-free' training regimes to eliminate the need for subwords entirely (Clark et al., 2022; Xue et al., 2022).

Another alternative input representation is to split words into morphemes, which provide theoretical benefits over subwords and have their own analytical and practical benefits particularly for morphologically rich languages (Üstün et al., 2018; Nzeyimana and Niyongabo Rubungo, 2022; Fan and Sun, 2023). Mapping orthographic text to morphemes continues to be a challenging task, requiring dedicated systems trained on labeled corpora (Batsuren et al., 2022) and we do not consider morphemes in this work.

2.3 Removal of Word Boundaries

When using a phonemic input representation to model speech, word boundaries are not typically included, as word boundaries are not explicitly marked in the speech stream. The phoneme stream representation (i.e., the combination of all three transformations) is the typical representation for word segmentation studies, where the task is to learn word boundaries without supervision (Brent, 1999). A wide variety of statistical, dynamic programming and neural approaches have been applied to the task, with consequences for acquisition research and low-resource language modeling (Blanchard et al., 2010; Çöltekin, 2017; Algayres et al., 2022; Goriely et al., 2023).

2.4 Input Representation Comparisons

To the best of our knowledge, a full systematic comparison of the three input transformations has not yet been conducted. Hahn and Baroni (2019) investigated the effect of removing word boundaries and using a word-level or character-level tokenization, evaluating on several psycholinguistic benchmarks. However, they only used graphemic text from Wikipedia and did not ablate the two transformations, only comparing a word-level model (with word boundaries) to a character-level model (without word boundaries). Nguyen et al. (2022) extend this work, comparing character-level graphemic input (with and without word boundaries) to character-level phonemic input (with and without word boundaries) by training on the Librispeech corpus (Panayotov et al., 2015). They also compare larger units of tokenization (BPE and word-level) for both graphemic and phonemic text, but only with word boundaries included, missing out on several key combinations.

In our work, we provide a complete comparison of these three input representation transformations by considering all combinations, leading to new input representations that have not been studied before (such as subword tokenization trained without word boundaries). We also use a larger model than previous work, a 12-layer transformer rather than a 3-layer LSTM.

3 Phoneme Stream Pipeline

To convert the data to a phonemic representation, we developed the **Corpus Phonemizer** tool:[1] a li-

[1] https://github.com/codebyzeb/Corpus-Phonemizer

brary to convert various corpora across many different languages to a unified phonemic representation in IPA, prepare them as Huggingface datasets and subsequently train Huggingface tokenizers.

3.1 Dataset Phonemization

Our toolkit leverages the `phonemizer` package (Bernard and Titeux, 2021) with the `espeak-ng` backend[2] which uses a combination of a pronunciation dictionary and pronunciation rules to convert orthographic transcriptions to IPA. We select the American English accent (`en-US`) for a consistent pronunciation.

The tool outputs phonemes separated by spaces.[3] For instance, the phonemic representation of "what a conundrum!" is:

w ʌ t ␣ ʌ ␣ k ə n ʌ n d ɹ ə m ␣

One limitation of our phonemization tool is that 'a' is not reduced to the shwah, 'ə' as it would be in continuous speech. We discuss the limitations of this phonemization process in section 6.2. Crucially, we lose punctuation marks, as they are an artefact of orthographic text and equivalent information in speech would be conveyed through prosody, stress, or non-linguistic signals such as gestures, none of which are included in this simple phonemic format. This has potential consequences for downstream tasks that rely on such markers, as discussed in section 5.3.

3.2 Tokenizer Preparation

Using the phonemic data transcribed by the Corpus Phonemizer tool, our pipeline then implements the three input transformations by preparing different tokenizers:

- Character tokenization We either train the tokenizer using the Byte-Pair Encoding (BPE) algorithm (Sennrich et al., 2016) (✗) or create a character-based tokenizer by extracting a vocabulary from the data (✓).
- Word boundary removal We either train the tokenizer with whitespace included (✗) or use the tokenizer's normalizer to strip whitespace (✓).
- Phonemic transcription The tokenizer is either trained on the original orthographic dataset (✗), or the phonemized version described above (✓).

These transformations can be made independently, allowing for all eight combinations of the transformations to be implemented as individual tokenizers. For the combination of BPE and no word boundaries, the whitespace is removed before training, so the model may learn 'subwords' that cross word boundaries.

Each tokenizer also adds a dedicated "utterance boundary" token `UTT_BOUNDARY` to the start of each sentence, representing the pauses between spoken utterances and serving as a dedicated start-of-sentence token. When sentences are collated, it also implicitly acts as an end-of-sentence token, as discussed in appendix B.2.

4 Experimental Setup

We evaluate the effect of our proposed input adjustments by training a GPT-2 model (Radford et al., 2019) using the BabyLM challenge framework (Choshen et al., 2024). The model is trained eight times with each combination of the three input adjustments. Following the STRICT track of the BabyLM challenge, we train on a provided corpus of 100 million words and evaluate on a series of benchmarks assessing the grammatical knowledge and the downstream capabilities of each model. We additionally evaluate on BabySLM (Lavechin et al., 2023) which provides syntactic and lexical scores specifically for speech-based models. Our phonemized dataset, trained models and tokenizers are hosted on Huggingface.[4]

4.1 Dataset

The BabyLM 2024 pretraining data contains 100 million words sourced from nine different corpora (Warstadt et al., 2023). Over 50% of the data consists of transcribed or scripted speech and over 40% comes from child-directed sources (written or spoken). We apply minor cleaning operations to the dataset, removing extraneous spaces and formatting anomalies using regular expressions.

4.2 Tokenizers

For each of the eight combinations of the three transformations, we train a tokenizer on the 'train' portion of the BabyLM dataset. We compare the

[2] `https://github.com/espeak-ng/espeak-ng`

[3] It is common practice to separate phonemes by spaces to make tokenization simple, as some individual phonemes may consist of several symbols, e.g. tʃ or ɜɪ.

[4] `https://huggingface.co/collections/phonemetransformers/from-babble-to-words-66e068b54765a48ff30273c9`

Model	Character tokenization	Word boundary removal	Phonemic transcription	Vocabulary Size	Example Tokenization	BLiMP Filtered	BLiMP Supplement	GLUE	BabySLM (Syntactic)	BabySLM (Lexical)
Baby Llama	✗	✗	✗	16,000	␣what ␣a ␣con und rum ␣!	73.1	60.6	69.0	94.0	-
LTG-BERT	✗	✗	✗	16,000	␣what ␣a ␣con und r um ␣!	69.3	66.5	68.4	75.8	-
GPT-2	✗	✗	✗	16,000	␣what ␣a ␣con und rum ␣!	**77.8**	**69.4**	**71.6**	92.8	-
	✗	✓	✗	16,000	what acon un drum !	73.9	64.3	68.6	73.9	-
	✗	✗	✓	16,000	␣wʌt ␣ʌ ␣kən ʌnd ɹəm	74.7	59.6	68.6	85.8	67.3
	✗	✓	✓	16,000	wʌt ʌkən ʌnd ɹəm	71.7	56.7	65.5	74.7	71.2
	✓	✗	✗	115	w h a t ␣ a ␣ c o n u n d r u m ␣ !	77.4	63.6	64.4	**94.9**	-
	✓	✓	✗	114	w h a t a c o n u n d r u m !	75.1	64.8	64.8	88.3	-
	✓	✗	✓	51	w ʌ t ␣ ʌ ␣ k ə n ʌ n d ɹ ə m	74.7	58.5	65.6	90.5	**89.6**
	✓	✓	✓	50	w ʌ t ʌ k ə n ʌ n d ɹ ə m	72.5	57.6	65.4	83.9	87.8

Table 1: Results for the two BabyLM baseline models and the GPT-2 model trained under all eight conditions. On the left, we compare the effects of each of the three transformations across all eight possible combinations, by tokenizing the example phrase "what a conundrum!". The '␣' character denotes word boundaries. On the right, we report BLiMP, GLUE and BabySLM scores achieved by each model, with the best scores in each column in **bold**.

output of the eight tokenizers in table 1. We used a vocabulary size of 16,000 for the BPE tokenizers to match the vocabulary size used by the two baseline models provided by the BabyLM challenge (described below).

Note that the vocabulary size for the character-level tokenizers operating on phonemes is less than half the vocabulary size of their orthographic equivalents. This is because the phonemic data only consists of the 47 phonemes produced by the American English accent, but the orthographic data includes numbers, punctuation and other symbols.

4.3 Model

Our experiments use the GPT-2 architecture. We train the model using all eight tokenizers (using the phonemized dataset for the phoneme-based tokenizers) for 400k steps, selecting the checkpoint with the lowest perplexity.[5] See appendix A for a full description of the chosen model parameters and training procedure.

We also report the results from two baseline models which achieved the highest scores at the 2023 BabyLM challenge. These are Baby Llama, an auto-regressive model, which was trained using knowledge distillation from an ensemble of teachers (Timiryasov and Tastet, 2023) and LTG-BERT, an architectural variation of the standard auto-encoding BERT architecture optimized for small, speech-based corpora (Samuel et al., 2023; Charpentier and Samuel, 2023). Both models use a BPE tokenizer with a vocabulary size of 16,000 and have a similar number of parameters to our model.[6]

4.4 Evaluation

We follow the BabyLM Challenge's framework and evaluate on BLiMP (Warstadt et al., 2020), BLiMP Supplement (Choshen et al., 2024) and a subset of the (Super)GLUE tasks (Wang et al., 2018, 2019). BLiMP assesses a model's ability to distinguish grammatical sentences from ungrammatical sentences across 67 subtasks covering a range of linguistic phenomena. BLiMP Supplement consists of 5 BLiMP-style tasks covering additional linguistic phenomena not tested by BLiMP. The GLUE suite assesses a language model's language understanding abilities on typical downstream tasks using fine-tuning.

We also evaluate our models on BabySLM (Lavechin et al., 2023), a benchmark specifically designed for probing speech-based LMs at a *syntactic* level and a *lexical* level. The benchmark was

[5]The best checkpoint for five of the eight models was the final checkpoint but a visual inspection of the curve revealed that differences between the final checkpoints were minimal.

[6]Our GPT-2 model has 85M non-embedding parameters. Baby Llama has 41M and LTG-Bert has 110M.

also designed to compare text-based models (those considered here, including both orthographic text and phonemic transcriptions) to speech-based models (which learn directly from audio) by providing parallel text and audio test instances. Finally, the vocabulary items were chosen to be compatible with children's language experiences, aiming to better reflect the input that children are exposed to as they begin to acquire language.

The BabySLM syntactic metric is similar to BLiMP, using pairs of grammatical and ungrammatical sentences, but consists of shorter sentences across just six simple syntactic phenomena. By comparison, BLiMP complicated many grammatical phenomena which may be rarely used even in adult–adult spontaneous conversation.

The lexical metric consists of minimal pairs of words and pseudo-words in a phonemic representation, representing a 'real-word recognition' task to assess a model's lexicon and phonemic capabilities. For instance, the model should assign a higher likelihood to the real-word t ɛ m p ɹ ə tʃ ə ɹ (temperature) compared to the pseudo-word t ɛ m p f ə tʃ ə ɹ (tempfature). This metric is related to the pronunciation of words, rather than the spelling of words and so cannot be used to evaluate models trained on orthographic text (which have no concept of pronunciation).

To evaluate our phoneme-based models, we run our phonemizer tool on all test instances across these benchmarks (except for the BabySLM lexical examples, which are already in IPA).

5 Results

In table 1, we report a summary of the results obtained by the two BabyLM baseline models and our GPT-2 model trained in all eight conditions. Due to limited computational resources we only train a single run per condition, limiting our ability to critique them individually. Exact results may be subject to variance across random seeds but we can still observe trends over the whole set.

The base GPT-2 model with no input adjustments outperforms the two baselines for BLiMP, BLiMP Supplement and GLUE, validating our selection of hyper-parameters and choice of architecture as described in appendix A.

Comparing the GPT-2 model with no input transformations (top row) to the same model with all three transformations applied (bottom row), we notice a decrease in performance across all benchmarks. Although this indicates that the GPT-2 architecture is best suited for the standard orthographic input representation (word boundaries, graphemes and subword tokenization), the decrease in performance when the three transformations are applied is not substantial and scores remaining competitive with the baseline models (all combinations still outperform LTG-BERT on BLiMP). It is clear that the model is still capable of learning grammatical rules and excelling at downstream tasks when the input consists of individual phonemes with no word boundaries.

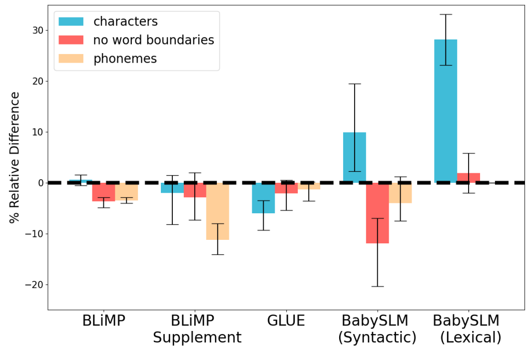


Figure 2: Mean (with Min and Max range) percentage difference achieved on each benchmark's macro score as a result of the three adjustments.

In section 5.1 we investigate this result further through an ablation of the three transformations, noting the effect of punctuation and context size. In section 5.2 we focus on the BabySLM metrics, which demonstrate a different pattern to the other benchmarks. Finally, in section 5.3 we investigate the consequences of removing punctuation in our phonemic transcriptions.

5.1 Teasing Apart the Three Transformations

By running our GPT-2 model with all eight combinations of the three input adjustments, we can tease apart the effect of each transformation.

For each transformation, we can create four pairs of runs that only differ with respect to that transformation (e.g. the four runs with a phonemic transcription and the four runs with orthographic text). For each pair, we calculate the percentage increase in each metric caused by the transformation. In fig. 2 we plot the average of these four percentage differences, allowing us to identify the overall effect of each transformation. We can also use the averaged scores for each subtask within a benchmark (such as the 67 BLiMP subtasks) to assess whether differences are significant for BLiMP, BLiMP Supplement, GLUE and BabySLM (Syntactic) using a paired t-test (see appendix B.1 for details and p-values for each test conducted).

Character Tokenization We find that character tokenization does not significantly decrease performance on BLiMP or BLiMP Supplement compared to subword tokenization. This validates previous work which found that despite the higher computation costs, character-based language models are just as capable of learning language (Al-Rfou et al., 2019; Hahn and Baroni, 2019). We do find a significant decrease for GLUE but this may be due to the fact that many of the finetuning examples for GLUE are very long and our model's context size is only 128 tokens, leading to severe truncation. As character-based tokenizers output more tokens for the same sentence than BPE tokenizers, this means that for many GLUE tasks, necessary information is lost.

Word boundary removal We find that removing word boundaries significantly decreases the BLiMP score, but the decreases for BLiMP Supplement and GLUE are not significant.[7] In their investigation, Nguyen et al. (2022) found a decrease of 7-8% on their own phonemic version of BLiMP when word boundaries were removed, but here we observe only an average decrease of 3.7%. As they only trained 3-layer LSTMs, it is possible that larger models like ours are required to overcome the loss of word boundaries.

Phonemic Transcription Finally, we find that using a phonemic transcription instead of the original written text significantly decreases performance on BLiMP and GLUE, although the percentage decreases are small (3.5% and 1.5% respectively). It also leads to the largest decrease of 11.3% for BLiMP Supplement. We discuss a possible explanation for this particular decrease in section 5.3.

5.2 BabySLM

Unlike the other benchmarks, our best BabySLM score is not achieved by the model trained with the standard orthographic input representation. Instead, the best syntactic score of 94.9 is achieved by the model that uses character-based tokenization (on written text, with word boundaries) and the best lexical score of 89.6 is achieved by the model that uses character-based tokenization for phonemes. It is also worth noting that, to the best of our knowledge, these are the best BabySLM scores to date (see appendix B.3 for a detailed comparison).

Examining the effect of each condition, we find that using a phonemic transcription on average reduces the syntactic score by 4.0%, which is in line with the other benchmarks discussed above. Unlike the other benchmarks, the character tokenization condition **always leads to an improvement** for both BabySLM scores: an average increase of 9.9% for the syntactic score and 23.9% for the lexical score. The sentences used for the syntactic test are all very short compared to the BLiMP sentences (4 words long on average) so a more fine-grained representation may be more useful. For the lexical test, where single words are compared that often only differ by a single phoneme, it seems more appropriate to use a character-based tokenization as the model needs to learn the distributional properties of individual phonemes, which may be lost in subword units.

The removal of word boundaries has a contrasting effect on the two scores. It reduces the syntactic score by 11.9% but increases the lexical score by 1.9%, the only benchmark where removing word boundaries is a positive change. However, the best individual lexical score was achieved by the model that did include word boundaries, suggesting that word boundaries are a helpful signal for a model learning to distinguish words from non-words, possibly because they help separate short sequences of phonemes that appear across word boundaries but not within words.

For the syntactic score, the worst scores are achieved by the models that learn subwords without word boundaries. For these models, the BPE algorithm is essentially acting as an unsupervised word segmentation algorithm learning to split entire sentences into useful units. With a vocabulary size of 16,000, it seems we learn units smaller than words (morpheme-sized units such as "un" in table 1) but also units that cross word boundaries (such as "acon" in table 1). The resulting implicit subword boundaries seem to have particular consequences when evaluating the shorter BabySLM sentences. Using the BPE algorithm in this way could be of interest for word segmentation studies.

5.3 The Effect of Punctuation

Punctuation is a feature of written text that is rarely included in phonemic transcriptions, as it does not typically change the way that words are pro-

[7] Since there are only 5 tasks for BLiMP Supplement it is difficult to get a p-value below 0.05.

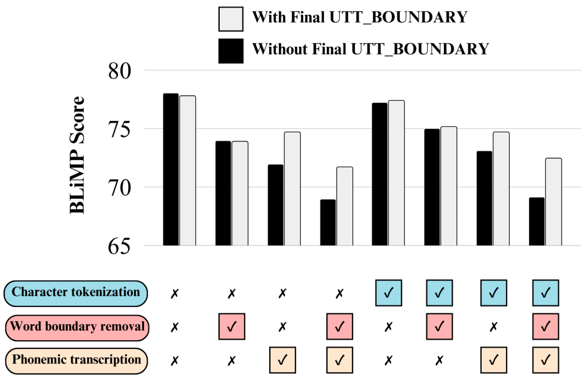


Figure 3: The overall BLiMP scores achieved by GPT-2 in our eight conditions with and without the `UTT_BOUNDARY` token (used to separate sentences) included at the end of evaluation instances.

nounced. However, punctuation in written text does carry important meaning about the structure and tone of sentences. In speech, this information is typically conveyed through intonation, stress and rhythm. By simply stripping punctuation in our phonemic transcriptions, we may be removing information that is important for a model's ability to learn and process language.

In some instances, naïvely stripping punctuation can even lead to nonsense sentences. This may explain the large dip in performance for BLiMP Supplement, as three of the five subtasks rely on punctuation to simulate question-answer pairs or dialogue, such as:

`A: What did you break?\nB: I broke a bowl.`

In the example above, the line break, colon and question mark are used to indicate speaker turns and convey the question-answer nature of the prompt. Removing the punctuation leads to a nonsense sentence, especially when read aloud with no pauses or change in tone to indicate the structure:

ʌ ␣ w ʌ t ␣ d ɪ d ␣ j uː ␣ b ɹ eɪ k ␣ b iː ␣ aɪ ␣ b ɹ oʊ k ␣ ʌ ␣ b oʊ l ␣

Without punctuation, the names "A" and "B" seem out of place. A model trained on written text can use punctuation to possibly understand that these are names, but a spoken model without punctuation would struggle to process this sentence.

This reliance on punctuation seems to be the leading cause of the drop in performance on BLiMP Supplement. If we remove the three subtasks where an understanding of punctuation is required to process the sentence, the effect of switching to a phonemic representation reduces the drop in performance considerably from 11.3% to 0.9%.

There is another subtle yet crucial consequence of removing punctuation: stripping punctuation at the end of sentences, if not handled correctly, can lead to significant decreases in performance on these benchmarks. This is because without an end-of-sentence marker, certain evaluation examples are no longer valid. In order to mark the end of the sentences without puncutation, we needed to ensure that our dedicated sentence-separation token was added to the end of each evaluation instance. The effect of this adjustment is highlighted in fig. 3. The increase in BLiMP score for our phonemic models confirms that this change was necessary and highlights the importance of carefully investigating the role of tokenization in the evaluation of large language models. We discuss this effect further in appendix B.2.

6 Discussion

In this work, we set out to establish whether modern language model architectures can encode grammatical knowledge and succeed at language understanding tasks when trained with phonemic input representations. By identifying three key transformations, carefully ablating them and evaluating our models on a wide variety of benchmarks, we found that these transformations do lead to decreased performance on standard benchmarks, but that this decrease is not substantial, and the effect of each transformation varies according to the evaluation. Generally, we conclude that language models are capable learners and training with these input representations is completely viable.

In this section, we consider explanations for the difference in performance across the benchmarks and discuss the limitations of phonemic transcriptions and our monolingual approach. Our work also has implications for human acquisition investigations and studies that train models directly from raw audio, which we discuss in appendix C.

6.1 The Effect of Input Transformations

There are many possible explanations for the decrease in performance for BLiMP, BLiMP Supplement and GLUE. In section 4.4 and section 5.3 we discuss two possibilities; the fact that character tokenization causes more substantial truncation (affecting GLUE) and the fact that phonemic transcriptions do not include punctuation (which particularly affects BLiMP Supplement). Another factor to consider is that although we do not change the

GPT-2 architecture or training parameters, the vocabulary size does change, which affects the size of the embedding layer. Character tokenization also leads to reduced exposure to each sentence during training (fewer epochs) because each sentence is represented with more tokens, increasing the number of steps required for each epoch. Furthermore, our initial choice of model parameters may have implicitly favored the standard orthographic input representation given that the language modeling community has been collectively optimizing these architectures to learn representations for written text, not phonemic streams. Just as the BabyLM challenge seeks to find solutions for low-resource language modeling, we may require an equivalent challenge to identify new methods and architectures for a phonemic input representation.

We also found a different pattern for the BabySLM benchmark, that certain transformations increased performance. In some cases, the transformations were even necessary (the lexical measure requiring a model to be trained on phonemic input). Given that the BabySLM benchmark more closely relates to child-language acquisition with its shorter sentences and vocabulary taken from child-directed speech, this result will be of interest to studies using language models to study acquisition.

6.2 Limitations and advantages of phonemic transcriptions

One difficulty in training models from ecological long-form child-centered audio is the lack of corpora available. Papers reporting research on day-long recordings tend not to release the raw data due to privacy concerns (e.g. Bergelson et al. (2023); Léon and Cristia (2024)). Our method allows us to convert text (which is much more readily available) into a speech representation (phoneme streams), meaning that we could quickly prepare a corpus of 100 million words.

There are also limitations in our transcription generation process. The fact that phonemes are an abstraction of speech means that we lose key information contained in speech such as prosody, stress and allophonic variation. Using a single accent to generate our phonemes, we also lose inter-speaker variability. Children also learn from non-linguistic cues, multi-modal input and interaction. If anything, it is a striking result that a model trained only on a set of 51 discrete symbols is able to demonstrate grammatical knowledge and perform competitively at downstream linguistic tasks.

6.3 Multi-lingual evaluation

A final important remark is that our experiments are conducted only in English. It is possible that language models trained on phonemic data in other languages would exhibit different trends in downstream performance. Although a multilingual analysis is outside the scope of our paper, we have applied our data processing pipeline to prepare phonemized datasets for 26 of the languages contained in the CHILDES database and hope to release this dataset in the near future.

7 Conclusion

Our study explores the effect of training language models using phonemic input representations, which offer both analytical and practical advantages. We develop a pipeline to convert orthographic datasets into a continuous stream of phonemes and leverage this pipeline to train a language model on phoneme streams and evaluate its grammatical and language understanding abilities. Our findings suggest that while phoneme-based input representations result in a slight decrease in model performance on traditional language understanding tasks, it is nonetheless a feasible training paradigm, facilitating future language modeling work, improving phonological interpretability and enhancing speech-based applications.

Acknowledgements

Our experiments were performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council. Zébulon Goriely's work is supported by The Cambridge Trust. Richard Diehl Martinez is supported by the Gates Cambridge Trust (grant OPP1144 from the Bill & Melinda Gates Foundation). Andrew Caines and Paula Buttery are supported by Cambridge University Press & Assessment. Lisa Beinborn's work is partially supported by the Dutch National Science Organisation (NWO) through the VENI program (Vl.Veni.211C.039).

References

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3159–3166.

Robin Algayres, Tristan Ricoul, Julien Karadayi, Hugo Laurençon, Salah Zaiem, Abdelrahman Mohamed, Benoît Sagot, and Emmanuel Dupoux. 2022. DP-Parse: Finding Word Boundaries from Raw Speech with an Instance Lexicon. *Transactions of the Association for Computational Linguistics*, 10:1051–1065.

Yamini Bansal, Behrooz Ghorbani, Ankush Garg, Biao Zhang, Colin Cherry, Behnam Neyshabur, and Orhan Firat. 2022. Data scaling laws in NMT: The effect of noise and architecture. In *International Conference on Machine Learning*, pages 1466–1482. PMLR.

Marco Baroni. 2022. On the proper role of linguistically oriented deep net analysis in linguistic theorising. In *Algebraic structures in natural language*, pages 1–16. CRC Press.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Lisa Beinborn and Nora Hollenstein. 2024. *Cognitive plausibility in natural language processing*. Springer.

Elika Bergelson, Melanie Soderstrom, Iris-Corinna Schwarz, Caroline F. Rowland, Nairán Ramírez-Esparza, Lisa R. Hamrick, Ellen Marklund, Marina Kalashnikova, Ava Guez, Marisa Casillas, Lucia Benetti, Petra van Alphen, and Alejandrina Cristia. 2023. Everyday language input and production in 1,001 children from six continents. *Proceedings of the National Academy of Sciences*, 120(52):e2300671120.

Mathieu Bernard and Hadrien Titeux. 2021. Phonemizer: Text to phones transcription for multiple languages in python. *Journal of Open Source Software*, 6(68):3958.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430.

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451.

Daniel Blanchard, Jeffrey Heinz, and Roberta Golinkoff. 2010. Modeling the contribution of phonotactic cues to the problem of word segmentation. *Journal of Child Language*, 37(3):487–511.

Benjamin Börschinger, Mark Johnson, and Katherine Demuth. 2013. A joint model of word segmentation and phonological variation for English word-final /t/-deletion. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1508–1516, Sofia, Bulgaria. Association for Computational Linguistics.

Michael R. Brent. 1999. Efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34(1):71–105.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.

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 BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2404.06214*.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Çağrı Çöltekin. 2017. Using Predictability for Lexical Segmentation. *Cognitive Science*, 41(7):1988–2021.

Shuangrui Ding, Zihan Liu, Xiaoyi Dong, Pan Zhang, Rui Qian, Conghui He, Dahua Lin, and Jiaqi Wang. 2024. Songcomposer: A large language model for lyric and melody composition in song generation. *arXiv preprint arXiv:2402.17645*.

Ewan Dunbar, Nicolas Hamilakis, and Emmanuel Dupoux. 2022. Self-Supervised Language Learning From Raw Audio: Lessons From the Zero Resource Speech Challenge. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1211–1226. Conference Name: IEEE Journal of Selected Topics in Signal Processing.

Emmanuel Dupoux. 2018. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59.

S Elizabeth Eden. 2018. *Measuring phonological distance between languages*. Ph.D. thesis, UCL (University College London).

Yanai Elazar, Akshita Bhagia, Ian Helgi Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, et al. 2024. What's in my big data? In *The Twelfth International Conference on Learning Representations*.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Linnea Evanson, Yair Lakretz, and Jean Rémi King. 2023. Language acquisition: do children and language models follow similar learning stages? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12205–12218, Toronto, Canada. Association for Computational Linguistics.

Allison Fan and Weiwei Sun. 2023. Constructivist tokenization for English. In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, pages 36–40, Washington, D.C. Association for Computational Linguistics.

Pablo Picasso Feliciano de Faria. 2019. The role of utterance boundaries and word frequencies for part-of-speech learning in Brazilian Portuguese through distributional analysis. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 152–159, Minneapolis, Minnesota. Association for Computational Linguistics.

Siyuan Feng, Ming Tu, Rui Xia, Chuanzeng Huang, and Yuxuan Wang. 2023. Language-universal phonetic representation in multilingual speech pretraining for low-resource speech recognition. In *INTERSPEECH 2023*, Dublin, Ireland. ISCA.

Robert Gale, Alexandra Salem, Gerasimos Fergadiotis, and Steven Bedrick. 2023. Mixed orthographic/phonemic language modeling: Beyond orthographically restricted transformers (BORT). In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 212–225, Toronto, Canada. Association for Computational Linguistics.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. 1993. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93:27403.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pages 517–520. IEEE Computer Society.

Zébulon Goriely, Andrew Caines, and Paula Buttery. 2023. Word segmentation from transcriptions of child-directed speech using lexical and sub-lexical cues. *Journal of Child Language*, pages 1–41.

Michael Hahn and Marco Baroni. 2019. Tabula nearly rasa: Probing the linguistic knowledge of character-level neural language models trained on unsegmented text. *Transactions of the Association for Computational Linguistics*, 7:467–484.

Mark Hasegawa-Johnson, Leanne Rolston, Camille Goudeseune, Gina-Anne Levow, and Katrin Kirchhoff. 2020. Grapheme-to-phoneme transduction for cross-language asr. In *International Conference on Statistical Language and Speech Processing*, pages 3–19. Springer.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Nora Hollenstein, Federico Pirovano, Ce Zhang, Lena Jäger, and Lisa Beinborn. 2021. Multilingual language models predict human reading behavior. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 106–123, Online. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *Preprint*, arXiv:1602.02410.

Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673.

Nina Kazanina, Jeffrey S Bowers, and William Idsardi. 2018. Phonemes: Lexical access and beyond. *Psychonomic bulletin & review*, 25(2):560–585.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander Rush. 2016. Character-aware neural language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Christo Kirov and Ryan Cotterell. 2018. Recurrent Neural Networks in Linguistic Theory: Revisiting Pinker and Prince (1988) and the Past Tense Debate. *Transactions of the Association for Computational Linguistics*, 6:651–665.

Marvin Lavechin, Maureen De Seyssel, Hadrien Titeux, Hervé Bredin, Guillaume Wisniewski, Alejandrina Cristia, and Emmanuel Dupoux. 2022. Can statistical learning bootstrap early language acquisition? a modeling investigation.

Marvin Lavechin, Yaya Sy, Hadrien Titeux, María Andrea Cruz Blandón, Okko Räsänen, Hervé Bredin, Emmanuel Dupoux, and Alejandrina Cristia. 2023. BabySLM: language-acquisition-friendly benchmark of self-supervised spoken language models. In *INTERSPEECH 2023*, pages 4588–4592, Dublin, Ireland. ISCA.

Colin Leong and Daniel Whitenack. 2022. Phone-ing it in: Towards flexible multi-modal language model training by phonetic representations of data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5306–5315, Dublin, Ireland. Association for Computational Linguistics.

Yinghao Aaron Li, Cong Han, Xilin Jiang, and Nima Mesgarani. 2023. Phoneme-level BERT for enhanced prosody of text-to-speech with grapheme predictions. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Mathilde Léon and Alejandrina Cristia. 2024. *Data Protection Handbook for Long-Form Recording Research: Navigating Data Protection Laws across the Globe*.

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. CharBERT: Character-aware pre-trained language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 39–50, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Brian MacWhinney and Catherine Snow. 1985. The Child Language Data Exchange System. *Journal of Child Language*, 12(2):271–295.

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

Yevgen Matusevych, Thomas Schatz, Herman Kamper, Naomi H Feldman, and Sharon Goldwater. 2023. Infant phonetic learning as perceptual space learning: A crosslinguistic evaluation of computational models. *Cognitive Science*, 47(7):e13314.

Connor Mayer. 2020. An algorithm for learning phonological classes from distributional similarity. *Phonology*, 37(1):91–131.

Tu Anh Nguyen, Maureen De Seyssel, Robin Algayres, Patricia Roze, Ewan Dunbar, and Emmanuel Dupoux. 2022. Are word boundaries useful for unsupervised language learning? *arXiv preprint arXiv:2210.02956*.

Antoine Nzeyimana and Andre Niyongabo Rubungo. 2022. KinyaBERT: a morphology-aware Kinyarwanda language model. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5347–5363, Dublin, Ireland. Association for Computational Linguistics.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32.

Alan Prince and Paul Smolensky. 1997. Optimality: From neural networks to universal grammar. *Science*, 275(5306):1604–1610.

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.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Thomas Schatz, Naomi H Feldman, Sharon Goldwater, Xuan-Nga Cao, and Emmanuel Dupoux. 2021. Early phonetic learning without phonetic categories: Insights from large-scale simulations on realistic input. *Proceedings of the National Academy of Sciences*, 118(7):e2001844118.

Mark S. Seidenberg and James L. McClelland. 1989. A distributed, developmental model of word recognition and naming. *Psychological Review*, 96:523–568.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Mukuntha Narayanan Sundararaman, Ayush Kumar, and Jithendra Vepa. 2021. PhonemeBERT: Joint Language Modelling of Phoneme Sequence and ASR Transcript. In *Proc. Interspeech 2021*, pages 3236–3240.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. PhonologyBench: Evaluating phonological skills of large language models. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

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. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

Ahmet Üstün, Murathan Kurfalı, and Burcu Can. 2018. Characters or morphemes: How to represent words? In *Proceedings of the Third Workshop on Representation Learning for NLP*, pages 144–153, Melbourne, Australia. 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. In *Advances in Neural Information Processing Systems*, volume 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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, 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Jian Zhu, Changbing Yang, Farhan Samir, and Jahurul Islam. 2024. The taste of IPA: Towards open-vocabulary keyword spotting and forced alignment in any language. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 750–772, Mexico City, Mexico. Association for Computational Linguistics.

A Implementation Details

We implement all experiments using the `PyTorch` framework (Paszke et al., 2019) and the `Transformers` library (Wolf et al., 2020).

A.1 Hardware Details

We use a server with one NVIDIA A100 80GB PCIe GPU, 32 CPUs, and 32 GB of RAM for all experiments. Below, we report a subset of the output of the *lscpu* command:

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         46 bits physical,
                       48 bits virtual
Byte Order:            Little Endian
CPU(s):                32
On-line CPU(s) list:   0-31
Vendor ID:             GenuineIntel
Model name:            Intel(R) Xeon(R)
                       Silver 4210R CPU
                       @ 2.40GHz
CPU family:            6
Model:                 85
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             8
Stepping:              7
BogoMIPS:              4800.11
```

A.2 Model Parameters and Training Procedure

Parameter	Value
Layers	12
Heads	12
Dropout	0.1
Embedding Size	768
Inner Size	3072
Max Example Length	128
Learning Rate	0.001
Optimizer	AdamW
Scheduler Type	Linear
Max Steps	400,000
Warm-up Steps	90,000
Per Device Batch Size	32

Table 2: Hyperparameter settings for training the GPT-2 architecture. Vocabulary size varies according to the tokenizer used, but all other parameters are constant across experiments. Where values are not reported, they may be assumed to be default values.

We describe the model and training parameters in table 2. The model parameters were chosen to match those of the Pythia-170M model from the Pythia suite (Biderman et al., 2023). The model has 85M non-embedding parameters and is also equivalent in size to GPT-Neo 125M and OPT-125M. The Pythia models use the GPTNeoX architecture which is slightly different to GPT-2. In initial experiments, we found that GPT-2 performed better on the benchmarks across all eight of our conditions.

Data is prepared into batches by first tokenizing the entire dataset, combining all tokens into one long vector, and then splitting the vector into chunks of 128 tokens. Only the very last example is padded, if required. At each step during training, random chunks are selected and combined into batches.

Checkpoints are taken every 50,000 steps during training. At each checkpoint, the perplexity is evaluated on the held-back evaluation set, and at the end of training the checkpoint with the lowest perplexity is returned as the best model.

B Evaluation Details

B.1 Significance Tests

It is difficult to determine whether the results for a given benchmark are significant given that we only train a single run for each of the eight conditions. Instead, we calculate the significance of a particular transformation by comparing the scores for each subtask of a benchmark. We average the scores achieved by the four models with a transformation applied and average the scores achieved by the four models without the transformation applied, giving us paired results for each subtask. We then use a paired student t-test to assess the significance of the transformation. We give the p-values for our significance tests in table 3.

Note that there are 67 subtasks for BLiMP, 5 for BLiMP Supplement, 9 for GLUE and 9 for BabySLM (Syntactic). With only 5 pairs for BLiMP Supplement, the test is under-powered and low p-values are unlikely. There are no subtasks for BabySLM (Lexical) so significance cannot be computed in the same way.

B.2 The Effect of End-of-Sentence Tokens

By default, our tokenizers add a special start-of-sentence token `UTT_BOUNDARY` to all sentences. This corresponds to the <s> token often used by tokenizers to help transformers with sentence-level processing, and also represents utterance boundaries, which unlike word boundaries are a clear cue present in speech and often included in word segmentation studies (Feliciano de Faria, 2019).

Since sentences are collated together during training, this means that these tokens also appear at

	BLiMP	BLiMP Supplement	GLUE	BabySLM (Syntactic)
orthographic vs. phonemic	**0.0001**	0.0780	**0.0149**	0.1884
word boundaries vs. no word boundaries	**0.0000**	0.1831	0.0813	**0.0118**
character tokenization vs. subword tokenization	0.5069	0.4832	**0.0010**	0.1500

Table 3: p-values from the paired student t-tests for each experiment. Significant results are given in **bold** using an alpha level of 0.05.

the end of every sentence, implicitly acting as end-of-sentence tokens. As a result, the model may use them to represent sentence-level information (especially given that these models are auto-regressive). However, in most evaluation tasks, sentences are presented individually (with padding) and so by default the tokenizer does not add this token to the end of sentences.

This has consequences for zero-shot evaluation tasks where the grammaticality of the sentence depends on the sentence being marked as complete, which is the case for several of the BLiMP subtasks. For instance, one subtask evaluates a model's understanding of filler-gap dependencies by presenting grammatical "wh"-phrases with "that"-phrases that are ungrammatical due to a missing dependency. An example is given in table 4 along with the tokens produced by two of our tokenizers. Crucially, our phonemic transcriptions do not include punctuation (see section 5.3) and for this task, without an end-of-sentence marker, the "ungrammatical" sentence is no longer ungrammatical, as it could just be incomplete.

This means that the subtask remained a valid test for our orthographic models (due to the inclusion of punctuation to mark the end of the sentence), but not the phonemic ones, since for the phonemic models both the "grammatical" and "ungrammatical" sentences could be considered grammatical. Since this task is not balanced, any preference for the word "that" over the "wh"-words would lead to the model consistently choosing the "that" sentences and achieving results below chance (which is 0.5 for all BLiMP tasks).

In our initial experiments we found that the models trained on phonemes achieved scores between 0.06 and 0.14 for this task whereas the orthographic models achieved scores between 0.35 and 0.53. We then added the `UTT_BOUNDARY` token to the end of every evaluation instance and found that the phonemic models could then achieve scores between 0.26 and 0.34 (with little change for the orthographic models). These results also held for several other BLiMP tasks with similar constructions.

We thus decided to ensure that the token was added to the end of every evaluation instance for all benchmarks reported in this paper for two reasons. First, it acts as a necessary end-of-sentence marker to ensure certain tests remain valid for the phonemic models, and second, because the token may encode useful sentence-level information for all models (particularly for GLUE tasks, as only the encoding of the final token is used for predictions).

We present the effect of this decision in fig. 3 which reports the overall BLiMP scores for our eight conditions with and without the inclusion of the `UTT_BOUNDARY` token at the end of each evaluation sentence. There is a very large increase for all four phonemic models with little change for the orthographic models, confirming how crucial this change was to make.

B.3 BabySLM Comparison

In table 1 we report the BabySLM scores achieved by our models and in section 5.2 we mention that these are the highest scores achieved on this benchmark to date. It is worth noting that this is only in comparison to the baseline scores released with the BabySLM benchmark (Lavechin et al., 2023), as at the time of writing no other scores have been published for this benchmark, given how recently it was introduced.

In their study, Lavechin et al. (2023) achieved their highest syntactic score of 70.4 using BabyBERTa (Huebner et al., 2021) trained on only 5 million words from CHILDES (MacWhinney and Snow, 1985). All of our models beat this score, with the highest achieving 94.9. BabyBERTa also uses a BPE tokenizer whereas we found that a character-based tokenizer consistently gave better performance (see section 5.2). There is also an architectural difference, BabyBERTa is an autoencoder trained using masked language modeling, whereas our model is autoregressive, using next-token prediction. The LTG-BERT baseline, which is a similarly sized model also trained on 100 million words, only achieves a score of 75.8. The Baby Llama baseline, by comparison, achieves 94.0. It is possible that the autoregressive architecture is much more suited to the syntactic task than the

	Grammatical	Ungrammatical
Original	Patrick revealed what a lot of men wore.	Patrick revealed that a lot of men wore.
BPE Text Tokenizer	<s> ␣patrick ␣revealed ␣what ␣a ␣lot ␣of ␣men ␣wore ␣.	<s> ␣patrick ␣revealed ␣that ␣a ␣lot ␣of ␣men ␣wore ␣.
BPE Phoneme Tokenizer	<s> ␣pætɹɪk ␣ɹɪviːld ␣wʌt ␣ʌ ␣lɑt ␣ʌv ␣mɛn ␣wɔɹ	<s> ␣pætɹɪk ␣ɹɪviːld ␣θæt ␣ʌ ␣lɑt ␣ʌv ␣mɛn ␣wɔɹ

Table 4: An example sentence pair from the `wh_vs_that_with_gap` subtask in BLiMP and the outputted tokens from our two tokenizers that use subwords but do not remove word boundaries. The '␣' character denotes word boundaries and the '<s>' token represents our `UTT_BOUNDARY` token which acts as an utterance boundary and a start-of-sentence token.

autoencoder architecture of BERT.

When it comes to the lexical test, the highest score achieved by Lavechin et al. (2023) was 75.4 using a 3-layer LSTM trained on 1.2 million words from the Providence corpus (Börschinger et al., 2013) which they converted to a stream of phonemes with no word boundaries using a similar tool to ours. Our highest-scoring model was also trained with character-based tokenization of phonemes, but did include word boundaries, achieving a score of 89.6. Our model without word boundaries got the second-highest score with 87.8.

In both cases, our model is larger (12 layers) and trained on much more data (100 million words) than the BabySLM baselines. Also, our pre-training dataset contains a wider variety of sentences than just the child-directed utterances in CHILDES. We are currently investigating the effect of model size and training size on the BabySLM scores. In initial experiments, we found that even a 6-layer model trained on only 7 million words from CHILDES was able to achieve a lexical score of 82, but this model also only achieved a syntactic score of 70. We hypothesize that lexical-level knowledge can be learned with less data and by smaller models when compared to learning syntactic knowledge, but this research is ongoing.

C Further Implications

C.1 Comparing Human Acquisition to Language Model Learning

The capacity of LMs to learn language from text alone has spurred interest in using such models for acquisition and psychology studies, such as comparing model learning trends to child learning behaviour (Evanson et al., 2023) and using model outputs to predict human reading times (Hollenstein et al., 2021).

To push this research further, recent efforts aim to make language modeling more cognitively plausible (Beinborn and Hollenstein, 2024) by reducing the advantages that typical language models have over humans during the learning process (Warstadt and Bowman, 2022). One approach is to limit and curate the dataset to that which a typical human may be exposed to, such as is done in the BabyLM challenge (Warstadt et al., 2023). Another approach is to use an input representation that more closely mimics speech rather than written text (Dupoux, 2018). Finally, we must consider whether the architectures themselves are suitable linguistic theories, given that they were developed for downstream tasks (Baroni, 2022).

In this work we contribute to all three approaches by training a language model with streams of phonemes and assess whether the language model architecture used is advantaged or disadvantaged by these changes according to a wide variety of benchmarks. We hope that this leads to further work studying acquisition using phoneme streams as an input representation. However, while streams of phonemes may seem more cognitively plausible than written text, many studies go further than we do and seek to train directly on raw audio.

C.2 Learning directly from audio

Our study focused on alternative input representations for text-based language models, but there is also a field of work dedicated to training models directly from raw audio. In recent years, the Zero Resource Speech Challenge has helped pioneer the development of models that learn unsupervised from raw audio (Dunbar et al., 2022). Models such as STELA (Schatz et al., 2021; Lavechin et al., 2022) use a two-stage approach, learning a discrete symbolic representation by clustering 10ms chunks of audio, then feeding these to a multi-layered LSTM language model.

These models are also used to study acquisition, regarding raw audio as an input representation that is more cognitively plausible than phonemes; a continuous signal full of noise and non-linguistic information that children must learn to filter. Whether adults even use phonemes as a core linguistic representation, and whether children learn phonemic categories before other stages of acquisition both continue to be a matter of debate (Kazanina et al., 2018; Matusevych et al., 2023) and the symbolic representations learned by models such as STELA have a duration four times shorter than phonemes, challenging the assumption that phonemic categories are precursors to later stages of acquisition.

The gap in linguistic performance between text-based models and audio-based models continues to be substantial. Lavechin et al. (2023) developed BabySLM to compare text-based models to speech-based models and highlighted this gap, but further noted that even speech-based models may not always train on plausible input, many often using audiobooks as their training data (Kahn et al., 2020). When training the STELA model on 1024 hours of ecological long-form child-centered audio compared to 1024 hours of audiobooks, Lavechin et al. (2023) found that the model trained on long-form audio achieved chance-level syntactic and lexical capabilities, highlighting how far we are from producing architectures that can learn from the same signals as human children.

Graphemes vs. phonemes: battling it out in character-based language models

Bastian Bunzeck, Daniel Duran, Leonie Schade and **Sina Zarrieß**
CRC 1646 – Linguistic Creativity in Communication
Department of Linguistics
Bielefeld University, Germany
`{firstname.lastname}@uni-bielefeld.de`

Abstract

We present `grapheme-llama` and `phoneme-llama`, character-based language models trained for the 2024 BabyLM challenge. Through these models, we explore an under-researched approach to downsizing: replacing subword-based tokenization with character-level tokenization, drastically reducing the vocabulary size. The grapheme model is trained on a standard BabyLM dataset, while the phoneme model uses a phoneme-converted version of this dataset. Results show that grapheme-based models perform better overall, achieving scores comparable to subword-based models on grammatical benchmarks. Despite lower performance, phoneme models also demonstrate promising grammatical learning. We argue that our results challenge conventional wisdom on language modeling techniques and open up novel research questions with character- and phoneme-based models as objects of inquiry.

1 Introduction

While *large* language models continue to beat benchmarks, their parameter numbers, amounts of training corpora and training FLOPs are ever increasing. More recently, however, a new research focus on ecologically friendly, data-efficient and possibly cognitively plausible language models – so called BabyLMs – has emerged. But what makes a language model a *BabyLM*? For the BabyLM challenges (Warstadt et al., 2023; Choshen et al., 2024), BabyLMs are defined by extremely constrained data settings. In this constrained data setting, the best scoring models in the 2023 challenge employed highly sophisticated and large-ish architectures: ELC-BERT (Charpentier and Samuel, 2023) used numerous architectural improvements over standard encoders, while BabyLlama (Timiryasov and Tastet, 2023) was distilled from various larger teacher models. Models with architectures downsized similarly to their training data (e.g. by Veysel Çağatan, 2023, Bunzeck and Zarrieß, 2023 or Fields et al., 2023) did not fare as well on standard benchmarks.

As our submission to the 2024 BabyLM challenge (Choshen et al., 2024), we present `grapheme-llama`[1] and `phoneme-llama`[2]. We replace the standard subword-based tokenization algorithms with naive character-based tokenization, leading to a drastic decrease in vocabulary size. We show that when such simplifications are combined with state-of-the-art architectures like Llama (Touvron et al., 2023b), the resulting models still achieve considerable grammatical proficiency and provide useful inductive biases for further fine-tuning. While the grapheme model is trained on the standard 100M BabyLM data, our phoneme model is trained on a version of this data set converted to phonemes[3]. Although it performs generally worse than its grapheme counterpart, the phoneme model still manages to learn the grammatical phenomena in a matched BLiMP data set quite well. In the light of these results, we offer some discussion points for phoneme-based language modeling, the pitfalls it is currently facing and its general potential. In sum, we argue that these results open fruitful avenues for further research on small language models and question “common wisdom” in current language modeling practices.

2 Related work

Small LMs/downsizing: Recently, there has been a surge in interest in small-ish language models. The arguably first BabyLM, BabyBERTa

[1] `https://huggingface.co/bbunzeck/grapheme-llama`

[2] `https://huggingface.co/bbunzeck/phoneme-llama`

[3] In line with the G2P literature (cf. Moore and Skidmore, 2019; Ashby et al., 2021), we use (i) the term “phoneme” loosely to refer to (symbols for) types of speech sounds and (ii) the term “grapheme” loosely to refer to the letters of orthographic alphabets.

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 54–64
November 15-16, 2024

(Huebner et al., 2021), followed a combined (i.e. data *and* architecture) downsizing approach and showed that dramatically less training data can result in remarkable linguistic proficiency with a small model architecture. On the other hand, current "small" models often employ more complex strategies to achieve compactness, e.g. distillation with teacher and student models (Timiryasov and Tastet, 2023), or reduction of number precision (Wang et al., 2023). These models' "smallness" is only achieved after complex training procedures. In contrast to these developments, the BabyLM 2023 submissions by Veysel Çağatan (2023), Bunzeck and Zarrieß (2023) and Fields et al. (2023) used *a priori* small models (in terms of parameter size) to show the lower bounds of knowledge learnability from small data. They all showed that very small models (even models with a parameter size below 1M) can achieve scores equal to much larger baselines on standard evaluation tasks like BLiMP or GLUE. As such, these successful experiments give impetus for our current models: against common wisdom, the reduction of certain models hyperparameters does not have to have a detrimental effect on performance (a fact also corroborated by Muckatira et al., 2024). Comparable studies have neither focused on character-level tokenization nor on phoneme-based representations (see paragraphs below for the most comparable studies available), so we pioneer into this uncharted territory with our models.

Character-level LMs: While research on LMs with character-level tokenization is not exactly scarce, they have yet to gain widespread adoption. Character-based models have been inplemented for different architectures: the CANINE (Clark et al., 2022) architecture is a character-level encoder, the ByT5 (Xue et al., 2022) models employ a T5 encoder-decoder architecture with a Byte-level tokenizer and the Charformer models (Tay et al., 2022) use a tokenization module (GBST) that learns latent subword representations from characters. For all three models it has been shown that their specific pre-training regimens do provide useful inductive biases for further fine-tuning and that such are more robust to character-level noise than regular subword-tokenization models. Moreover, phonological categories like consonants and vowels are retrievable from CANINE (see Agirrezabal et al., 2023) – properties of language that are by design not captured by coarse-grained subword representations. From a more application-driven standpoint, El Boukkouri et al. (2020) have shown that character-level modeling can improve performance in the medical domain. Finally, Edman and Bylinina (2023) showed in the context of last year's BabyLM challenge that first training on a character-level vocabulary and then expanding it to the subword-level provides mixed effects on model performance, depending on the context size. It should also be noted that there are further approaches to language modeling without complex tokenization algorithms: Rust et al. (2023) show that LMs trained on pixel-based representations can help LMs excel at various syntactic and semantic tasks in typologically diverse languages, including non-Latin scripts.

Phoneme LMs: So far, phoneme-based LMs have mostly been trained as encoders to provide inductive biases for further fine-tuning on downstream tasks. PhonemeBERT (Sundararaman et al., 2021), Mixed-Phoneme BERT (Zhang et al., 2022) and XPhoneBERT (Nguyen et al., 2023) are examples for such models, which have been reported to improve downstream performance on various tasks, e.g. on text-to-speech. In contrast, the CharsiuG2P model (Zhu et al., 2022) is an encoder-decoder architecture explicitly pre-trained for grapheme-to-phoneme conversion (G2P). Purely autoregressive phoneme models have not received scientific attention, yet.

3 Methodology

3.1 Data

We train our models on the 100M BabyLM 2024 data set. This data set contains both (transcribed) spoken and written language. It includes spoken language from CHILDES (MacWhinney, 2000), the BNC (Burnard, 2007), Switchboard (Stolcke et al., 2000) and OpenSubtitles (Lison and Tiedemann, 2016), and written language from children's books in Project Gutenberg (Gerlach and Font-Clos, 2020) as well as a portion of the Simple English Wikipedia. Because the raw data contains extensive metadata and markup, we used an expanded version of the cleaning script from Timiryasov and Tastet (2023) to clean the data.

For our phoneme-based models, we then convert the cleaned data from graphemes to phonemes – a mapping from orthographic letters to sound-symbols to represent the pronunciation of the text. To convert our text to IPA (International Phonetic Association, 1999) symbols, we use the rule-based G_i2P_i system for G2P-conversion (Pine et al., 2022)[4], expanded by a manual replacement list that we compiled for contractions that this tool does not handle well. As the authors report no G2P accuracy for English, we conduct a manual evaluation on three short texts. We find a word-error-rate of 5.8% (tokens=363, errors=21), which we deem as sufficient for the sake of the current paper. For evaluation purposes, we also perform the same G2P conversion on the BLiMP data. We make this data set[5] and our converted training data[6] available on the Hugging Face hub.

3.2 Training

We use the `transformers` library (Wolf et al., 2020) to train four small, character-level llama models (Touvron et al., 2023b). All our models share equivalent model internals and training hyperparameters:

- Training tokens: 100M
- Hidden layers: 8
- Attention heads: 8
- Embedding size: 512
- Context size: 64
- Number of parameters: 15M/14.9M (grapheme-based/phoneme-based models)

We train two models on the original grapheme-based BabyLM data and two models on our converted phoneme-based data: for each data regimen, one model with whitespaces separating lexical tokens and one without these whitespaces. As we experiment with removing information about words by not using sub-word tokenization, the models without whitespaces can be seen as more extreme variants of the same training setting – they have (apart from beginning and end of sequences) no access to word segmentation information at all. To force the models to use more local information, we restrict the context size to 64 tokens (although we acknowledge that this might lead to detrimental performance on tasks that require longer contexts, especially EWoK and GLUE).

To implement character-level language modeling, we modify the tokenizers used for our models. Instead of the standard BPE tokenization algorithm, we simply fill our tokenizers' vocabularies with all unique characters in the respective pre-training corpora. For the grapheme-based models, this adds up to a vocabulary size of approx. 360. For the phoneme models, the vocabulary size is approx. 260. Next to the standard ASCII and IPA characters, these vocabularies are still so "large" due to a number of emojis and other non-linguistic Unicode characters included. Because some IPA symbols are also ordinary letters of Latin alphabets, and also due to the aforementioned non-alphabetic symbols, the vocabularies of the models share 118 tokens.

As training hyperparameters, we chose a batch size of 16, 200 warmup steps, and a learning rate set to 3e-4 in accordance with Touvron et al. (2023a). We train our models for five epochs, equaling roughly 25–28 hours of per-model training time on a single NVIDIA RTX A4000 GPU.

3.3 Model evaluation

In line with the BabyLM challenge, we evaluate our models through the BabyLM evaluation pipeline (Choshen et al., 2024; Gao et al., 2023). It includes three tasks – BLiMP (Warstadt et al., 2020), EWoK (Ivanova et al., 2024) and (Super)GLUE (Wang et al., 2018, 2019).

BLiMP is a collection of minimal pairs (ungrammatical vs. grammatical sentences) for English, including mostly (morpho)syntactic phenomena, but also semantic and (in the supplementary data) discourse-pragmatic minimal pairs. Although it suffers from a few shortcomings (partially nonsensical sentences, cf. Vazquez Martinez et al., 2023; a too restrictive binary notion of grammaticality that does not allow creative language use, etc.), it is a valuable resource and basically *the* linguistic benchmark for the evaluation of language models. If a model consistently manages to score the grammatical sentence as more plausible (i.e. through lower perplexity) it is said to have mastered the corresponding phenomenon. We evaluate all of our models on the regular BLiMP, and additionally on a matched BLiMP that contains the BLiMP data converted to match the data set the respective model was trained on (grapheme/phoneme, whitespace/no whitespace).

EWoK (Ivanova et al., 2024) is a benchmark that is supposed to measure world knowledge by testing models on their ability to match target texts with plausible/implausible contexts. It covers domains such as material properties, physical dynamics or

[4] We also tried a neural system (Zhu et al., 2022), but found it to be much less performant and of slightly worse transcription quality.

[5] https://huggingface.co/datasets/bbunzeck/phoneme-blimp

[6] https://huggingface.co/datasets/bbunzeck/phoneme-babylm-100M

social interactions. The sentence pairs function as minimal pairs (of pairs) and can therefore be evaluated in the same way as BLiMP examples. As both our grapheme models and the BabyLM baselines do not perform above chance on this benchmark, we decided not to create a phoneme version.

The (Super)GLUE tasks (Wang et al., 2018, 2019) are focused on more fine-grained language understanding and involve additional fine-tuning on task examples. As such, they measure how well our pre-training procedure supplies our models with useful inductive biases for the acquisition of these reasoning tasks, e.g. textual entailment or sentiment prediction. For reasons of time and resources, we opted to do parameter-efficient fine-tuning with LoRA (Hu et al., 2022) instead of full fine-tuning runs. In contrast to the provided fine-tuning script, we opted for only 16 epochs and a larger learning rate of 5e-4, in hopes to help our models converge faster. Due to a technical problem (and lack of time), we could only run one fine-tuning epoch for the MNLI sub-task. We also opted to not create a phonemized (Super)GLUE data set, for the same reasons as for EWoK.

4 Results

4.1 Zero-shot

The BLiMP results are collected in Table 1. With regard to the standard grapheme and whitespace BLiMP, the corresponding grapheme model also performs best. With a score of almost 72%, our character-based grapheme model is close to the subtoken-based autoregressive baseline (BabyLlama, 73.1%), and beats the masked LM baseline (LTG-BERT, 69.2%; not listed in Table 1). While the model trained without whitespace performs worse, the score of 59.88% is still far above chance. The phoneme models, on the other hand, only achieve scores that oscillate somewhat around the chance baseline. This is not surprising, as the overlap in vocabulary between the grapheme and phoneme models is small – the phoneme models can hardly retrieve any useful information from grapheme input. On the BLiMP supplement, none of our models achieve a score significantly higher than the chance baseline.

When considering the matched BLiMP evaluations, where we preprocess the BLiMP data in the same way as the pre-training corpus data, we can report much higher BLiMP scores. All four models perform way above chance, although both the G2P conversion and the deletion of all whitespace have a detrimental effect on the scores. Interestingly, the grapheme model without whitespaces achieves the best score on the BLiMP supplement (56.28%), although we can only speculate as to why (see Discussion for an attempt at explanation).

This picture gets even more complicated when we consider the individual BLiMP paradigms. The full BLiMP scores for the matched evaluation can be found in Appendix A. While the grapheme whitespace model generally performs best across the most paradigms, each model still features some high scores. For certain, highly-specific phenomena (e.g. `sentential_negation_npi_scope_filtered`), the non-whitespace phoneme model – our overall weakest model – outperforms all other models. It remains open to further inquiry whether these scores are only training noise or caused by specific linguistic factors only instantiated by this specific combination of data preprocessing steps.

The evaluation results for EWoK (Table 2) display a very uniform picture. No model achieves any considerable score above the chance baseline for any phenomenon. This is also in line with the results of the baseline models, which seemingly do not learn any "world knowledge", as measured by EWoK.

4.2 Fine-tuning

The (Super)GLUE scores can be found in Table 3. They follow no clear pattern. While the average scores for the models are rather similar (and all fairly low in comparison to the baselines, like 63.3% for BabyLlama), the scores for the individual tasks are highly varied. While the standard grapheme model achieves the highest scores on six out of eleven included tasks, all other models also get at least one highest score. Averaged across all tasks, the grapheme model without whitespace is even better than its normal counterpart. The differences between models are immense and no structured conclusions about presumed effects of any variable (grapheme/phoneme, whitespace/no whitespace) can be drawn. It is especially surprising that the phoneme models, which do not contain the full grapheme-model vocabulary and therefore sometimes lead to somewhat corrupted/distorted tokenized versions of the data (e.g. through missing tokens), still seem to impart quite useful inductive biases for many of the included sub-tasks in (Super)GLUE: Only for CoLA, MNLI and MNLI-

BLiMP version	Grapheme model	Grapheme model, no whitesp.	Phoneme model	Phoneme model, no whitesp.	BabyLlama
BLiMP	**71.69%**	59.88%	44.05%	54.02%	73.1%
BLiMP supplement	52.30%	50.12%	**55.04%**	44.47%	60.6%
Matched BLiMP	**71.69%**	68.88%	66.90%	64.88%	73.1%
Matched BLiMP supplement	52.30%	**56.28%**	55.42%	54.13%	60.6%

Table 1: BLiMP accuracies for our four models and BabyLlama baseline (random baseline = 50%)

EWoK subtask	Grapheme model	Grapheme model, no whitesp.	Phoneme model	Phoneme model, no whitesp.	BabyLlama
agent-properties	49.46%	49.68%	**50.23%**	50.05%	-
material-dynamics	49.22%	49.61%	**49.87%**	48.87%	-
material-properties	48.24%	50.00%	50.00%	**50.59%**	-
physical-dynamics	48.33%	**51.67%**	50.00%	50.00%	-
physical-interactions	47.84%	50.18%	50.18%	**51.44%**	-
physical-relations	50.73%	49.14%	49.63%	**51.22%**	-
quantitative-properties	50.96%	**52.55%**	49.36%	49.04%	-
social-interactions	49.66%	50.34%	**51.02%**	**51.02%**	-
social-properties	**51.52%**	48.78%	50.30%	48.17%	-
social-relations	49.68%	49.29%	**50.00%**	**50.00%**	-
spatial-relations	46.73%	46.33%	**51.43%**	50.20%	-
Average	49.30%	49.80%	**50.20%**	50.10%	52.1%

Table 2: EWoK accuracies for our four models and BabyLlama baseline (random baseline = 50%)

mm, the scores achieved by the (in theory unfitting) phoneme models are close or equal to the random chance baseline. For the other tasks, especially SST2 and MRPC, scores are well above chance. Here, it remains questionable whether the inductive biases of our phoneme models actually affect the performance on (Super)GLUE, or if the whole fine-tuning process equals the adoption of some heuristic shortcuts to solve the problems tested by (Super)GLUE (see Gururangan et al., 2018; Belinkov et al., 2019 for discussions of artifacts in NLI data), to which only CoLA, MNLI and MNLI-mm are robust enough to resist.

5 Discussion

General remarks: There are two commonly presented arguments against character-level tokenization (e.g. presented in Clark et al., 2022): (i) such models achieve subpar results on evaluations; and (ii) as the computational complexity of a transformer grows quadratically with the input size, the token increase yields inefficient models. To (i) we can only reply that our results speak for themselves. The strong performance of such a small Llama model on BLiMP shows that character-based models are able to learn the structure of a language as well as its subword-based sister models. The comparatively lower performance on fine-tuning tasks is likely caused by the small architecture, and could be improved with more parameters. Also, the small context size of our models might be a limiting factor for the fine-tuning tasks (and also the zero-shot EWoK evaluation, as it contains fairly long sentences). To (ii) we can reply that this is not such a big concern, as we use small models and small-ish context sizes, anyway. While this approach might not be sufficient for models with billions of parameters, it surely is for BabyLMs.

Graphemes vs. phonemes: The comparison between our grapheme and phoneme models undoubtedly concludes with a win for the grapheme models. Across all benchmarks, they outperform the phoneme models on average. No clear tendencies spring to mind when analyzing the detailed results – however, all four models achieve best scores on some sub-tasks in benchmarks. Separating noise from signal in these results remains an open task for future studies. As of now, we can only speculate why the phoneme models perform *this* worse. An easy explanation could be the absence of punctuation in phoneme models. As dots, commas and other punctuation marks perform important semantic functions in texts (see Crystal, 2015), their absence quite possibly has a negative effect on the acquired grammatical system of a language model.

Another problem could lie in the quality of our G2P system. Alphabetic writing systems generally associate letters to sounds, and vice versa. However, especially for English, the correspondences between *graphemes* and *phonemes* are not trivial and (can seem) arbitrary (Pulgram, 1951; Venezky, 1967; Emerson, 1997; Roca, 2016). Graphemes are arranged according to orthographic conventions which usually do not directly reflect a language's underlying phonological system. Grapheme-to-

GLUE subtask	Grapheme model	Grapheme model, no whitesp.	Phoneme model	Phoneme model, no whitesp.	BabyLlama
CoLA (MCC)	**0.098**	0.0668	0.0325	0	-
SST-2	**74.31%**	74.08%	69.27%	72.94%	-
MRPC (F1)	79.75%	80.62%	81.05%	**81.29%**	-
QQP (F1)	66.54%	**71.04%**	62.40%	59.57%	-
MNLI	**52.59%**	50.15%	46.92%	45.60%	-
MNLI-mm	**51.32%**	50.24%	47.40%	46.30%	-
QNLI	59.26%	**63.84%**	55.01%	52.82%	-
RTE	44.60%	43.17%	51.08%	**58.27%**	-
BoolQ	64.46%	64.65%	**64.89%**	63.85%	-
MultiRC	**57.63%**	56.23%	57.26%	57.59%	-
WSC	61.54%	61.54%	59.62%	**62.46%**	-
Average	56.50%	**56.60%**	54.40%	54.70%	69.0%

Table 3: (Super)GLUE results for our models and BabyLlama baseline

phoneme conversion, as the computational attempt to solve this problem, cannot be considered as solved. Relatively high error rates of G2P tools are still an issue in speech and language processing. For example, the SIGMORPHON shared tasks on "multilingual grapheme-to-phoneme conversion" (Gorman et al., 2020; Ashby et al., 2021; McCarthy et al., 2023) use the metrics *word error rate* (WER) and *phone error rate* (PER) for evaluation. Word error rates of the best submissions in 2020 range from 24.89 (for Georgian) to 0.89 (for Vietnamese) (Gorman et al., 2020). As such, it might be more sensible to train on manually transcribed speech. Unfortunately, such corpora are small and rare, although it might be interesting to see whether some variation in phoneme data can influence performance on standard benchmarks.

Additionally, it remains questionable how phoneme data should be represented for language modeling. Splitting a transcription into a sequence of characters for character-level tokenization introduces some issues: Unicode defines IPA base symbols as individual characters. Some diacritics (which add information on fine phonetic detail to base symbols) are defined as "Spacing Modifier Letters", others as "Combining Diacritical Marks". Thus an aspirated alveolar plosive [t^{h}] or a long vowel [aː] are treated as two characters, while, depending on the treatment of composed Unicode characters, a de-voiced alveolar fricative [z̥] or a raised vowel [a̝] may be treated as one. Affricates (combined sounds), for example, may be represented as a sequence of two characters joined by a double diacritic [d͜ʒ], or as a single ligature [ʤ].

Whitespace: Finally, the detrimental effect of whitespace removal also deserves explanation and discussion. Whitespace encodes important linguistic information about word boundaries (or approximations thereof) -- information which is not available in spoken language (there, pauses between stretches of connected speech serve different purposes). Instead, prosody (e.g. word stress or intonation), provides cues to segmentation at different levels of linguistic abstraction (like words and phrases). This is, apart from whitespace, not reflected in orthographic texts and also often missing from phonetic transcriptions[7]. As such, data without whitespaces is a developmentally/cognitively/linguistically more plausible form of input. As this added plausibility comes with the loss of information, it is not surprising that scores for non-whitespace models are generally lower. A notable exception is the high score of the non-whitespace grapheme model for the matched BLiMP supplement. This might be a side effect of our very small context size. The BLiMP supplement contains inter alia dialogue phenomena with long dependencies. The models without whitespace can take in more (non-whitespace) characters, and in the light of our rather small context size, it might be the case that the whitespace models cannot process enough information to actually grasp these phenomena.

6 Conclusion

This paper has shown two things: (i) character-based tokenization is a viable alternative for small language models and (ii) phoneme-based LMs can also perform reasonably well on common benchmarks, although grapheme models are superior. With the drawbacks (e.g. the computational complexity increase in large models) of character-based tokenization, we of course do not want to replace sub-word tokenization. However, we believe that our models deserve a place in the toolbox of developmentally more plausible language models. They can be used to test what kind of linguistic knowledge

[7] Our phoneme data does not include word stress.

can be learned from raw input and answer questions about the learnability of linguistic knowledge from an even poorer stimulus (Thomas, 2002; Berwick et al., 2011) than the “stimulus” of subword models. In combination with phoneme representations, they open up new avenues of inquiry, e.g. for phenomena on the phonological/phonetic or lexical levels of linguistic analysis – phenomena which are not captured by the coarse-grained structure of subword tokens. Moreover, character-based language models open new pathways into experiments with multilingual models. The Latin script, for example, offers a shared vocabulary for many languages, whereas the IPA even offers a shared vocabulary for practically all languages.

Limitations

As previously mentioned, our results are only snapshots of individual training runs. Repeated training efforts with different initialization would be needed to filter noise from actual tendencies.

Besides, in the light of the current BabyLM challenge, we could only test these phenomena for English. The differences between grapheme and phoneme models may not generalize to other languages with different writing systems, languages with different levels of phonemic correspondences and systematicity in their orthography (like English or French vs Spanish or Czech), and languages with different morpho-phonological systems.

Acknowledgements

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – CRC-1646, project number 512393437, project A02.

References

Manex Agirrezabal, Sidsel Boldsen, and Nora Hollenstein. 2023. The Hidden Folk: Linguistic Properties Encoded in Multilingual Contextual Character Representations. In *Proceedings of the Workshop on Computation and Written Language (CAWL 2023)*, pages 6–13, Toronto, Canada. Association for Computational Linguistics.

Lucas F.E. Ashby, Travis M. Bartley, Simon Clematide, Luca Del Signore, Cameron Gibson, Kyle Gorman, Yeonju Lee-Sikka, Peter Makarov, Aidan Malanoski, Sean Miller, Omar Ortiz, Reuben Raff, Arundhati Sengupta, Bora Seo, Yulia Spektor, and Winnie Yan. 2021. Results of the second SIGMORPHON shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 115–125, Online. Association for Computational Linguistics.

Yonatan Belinkov, Adam Poliak, Stuart Shieber, Benjamin Van Durme, and Alexander Rush. 2019. Don't take the premise for granted: Mitigating artifacts in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 877–891, Florence, Italy. Association for Computational Linguistics.

Robert C. Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. 2011. Poverty of the Stimulus Revisited. *Cognitive Science*, 35(7):1207–1242.

Bastian Bunzeck and Sina Zarrieß. 2023. GPT-wee: How small can a small language model really get? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 7–18, Singapore. Association for Computational Linguistics.

Lou Burnard. 2007. Reference Guide for the British National Corpus (XML Edition).

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every Layer Counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 210–224, Singapore. Association for Computational Linguistics.

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. *Preprint*, arXiv:2404.06214.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an Efficient Tokenization-Free Encoder for Language Representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

David Crystal. 2015. *Making a Point: The Persnickety Story of English Punctuation*, 1 edition. St. Martin's Press, New York.

Lukas Edman and Lisa Bylinina. 2023. Too Much Information: Keeping Training Simple for BabyLMs. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 61–69, Singapore. Association for Computational Linguistics.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for Word-Level Open-Vocabulary Representations From Characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6903–6915, Barcelona, Spain (Online).

International Committee on Computational Linguistics.

Ralph H. Emerson. 1997. English spelling and its relation to sound. *American Speech*, 72(3):260–288.

Clayton Fields, Osama Natouf, Andrew McMains, Catherine Henry, and Casey Kennington. 2023. Tiny Language Models Enriched with Multimodal Knowledge from Multiplex Networks. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 19–29, Singapore. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Martin Gerlach and Francesc Font-Clos. 2020. A Standardized Project Gutenberg Corpus for Statistical Analysis of Natural Language and Quantitative Linguistics. *Entropy*, 22(1):126.

Kyle Gorman, Lucas F.E. Ashby, Aaron Goyzueta, Arya McCarthy, Shijie Wu, and Daniel You. 2020. The SIGMORPHON 2020 shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 40–50, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning More Grammar With Small-Scale Child-Directed Language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

International Phonetic Association, editor. 1999. *The Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press.

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 Akyürek, 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*.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Brian MacWhinney. 2000. *The CHILDES Project: Tools for Analyzing Talk*, 3 edition. Lawrence Erlbaum Associates, Mahwah, NJ.

Arya D. McCarthy, Jackson L. Lee, Alexandra DeLucia, Travis Bartley, Milind Agarwal, Lucas F.E. Ashby, Luca Del Signore, Cameron Gibson, Reuben Raff, and Winston Wu. 2023. The SIGMORPHON 2022 shared task on cross-lingual and low-resource grapheme-to-phoneme conversion. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 230–238, Toronto, Canada. Association for Computational Linguistics.

Roger K. Moore and Lucy Skidmore. 2019. On the use/misuse of the term 'phoneme'. In *Interspeech 2019*, pages 2340–2344. ISCA.

Sherin Muckatira, Vijeta Deshpande, Vladislav Lialin, and Anna Rumshisky. 2024. Emergent Abilities in Reduced-Scale Generative Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1242–1257, Mexico City, Mexico. Association for Computational Linguistics.

Linh The Nguyen, Thinh Pham, and Dat Quoc Nguyen. 2023. XPhoneBERT: A Pre-trained Multilingual Model for Phoneme Representations for Text-to-Speech. *Preprint*, arXiv:2305.19709.

Aidan Pine, Patrick William Littell, Eric Joanis, David Huggins-Daines, Christopher Cox, Fineen Davis, Eddie Antonio Santos, Shankhalika Srikanth, Delasie Torkornoo, and Sabrina Yu. 2022. Gi22Pi Rule-based, index-preserving grapheme-to-phoneme transformations Rule-based, index-preserving grapheme-to-phoneme transformations. In *Proceedings of the Fifth Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 52–60, Dublin, Ireland. Association for Computational Linguistics.

Ernst Pulgram. 1951. Phoneme and grapheme: A parallel. *WORD*, 7(1):15–20.

Iggy Roca. 2016. Phonology and English spelling. In Vivian Cook and Des Ryan, editors, *The Routledge Handbook of the English Writing System*, 1 edition, pages 65–91. Routledge, London.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. *Computational Linguistics*, 26(3):339–373.

Mukuntha Narayanan Sundararaman, Ayush Kumar, and Jithendra Vepa. 2021. PhonemeBERT: Joint Language Modelling of Phoneme Sequence and ASR Transcript. In *Interspeech 2021*, pages 3236–3240. ISCA.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2022. Charformer: Fast character transformers via gradient-based subword tokenization. In *International Conference on Learning Representations*.

Margaret Thomas. 2002. Development of the concept of "the poverty of the stimulus". *The Linguistic Review*, 19(1-2):51–71.

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. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 251–261, Singapore. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and Efficient Foundation Language Models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. LLaMA 2: Open Foundation and Fine-Tuned Chat Models. *Preprint*, arXiv:2307.09288.

Hector Vazquez Martinez, Annika Lea Heuser, Charles Yang, and Jordan Kodner. 2023. Evaluating Neural Language Models as Cognitive Models of Language Acquisition. In *Proceedings of the 1st GenBench Workshop on (Benchmarking) Generalisation in NLP*, pages 48–64, Singapore. Association for Computational Linguistics.

Richard L. Venezky. 1967. English orthography: Its graphical structure and its relation to sound. *Reading Research Quarterly*, 2(3):75–105.

Ömer Veysel Çağatan. 2023. ToddlerBERTa: Exploiting BabyBERTa for Grammar Learning and Language Understanding. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 143–151, Singapore. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. 2023. BitNet: Scaling 1-bit Transformers for Large Language Models. *Preprint*, arXiv:2310.11453.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Gotlieb 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 1–6, 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Guangyan Zhang, Kaitao Song, Xu Tan, Daxin Tan, Yuzi Yan, Yanqing Liu, Gang Wang, Wei Zhou, Tao Qin, Tan Lee, and Sheng Zhao. 2022. Mixed-Phoneme BERT: Improving BERT with Mixed Phoneme and Sup-Phoneme Representations for Text to Speech. In *Interspeech 2022*, pages 456–460. ISCA.

Jian Zhu, Cong Zhang, and David Jurgens. 2022. ByT5 model for massively multilingual grapheme-to-phoneme conversion. In *Interspeech 2022*, pages 446–450. ISCA.

A Full BLiMP scores

Phenomenon	Graph. model	Graph. model, no whitesp.	Phon. model	Phon. model, no whitesp.
BLiMP	**71.69%**	68.88%	66.90%	64.88%
BLiMP supplement	52.30%	**56.28%**	55.42%	54.13%
adjunct_island_filtered	73.17%	**76.72%**	35.24%	36.75%
anaphor_gender_agreement_filtered	85.48%	82.29%	**86.30%**	69.10%
anaphor_number_agreement_filtered	**97.10%**	88.51%	95.17%	87.00%
animate_subject_passive_filtered	68.60%	**71.62%**	68.83%	62.91%
animate_subject_trans_filtered	**91.01%**	90.57%	82.23%	77.79%
causative_filtered	**69.07%**	68.09%	66.01%	64.55%
complex_NP_island_filtered	43.38%	**47.28%**	38.30%	43.85%
coordinate_structure_constraint_complex_left_branch_filtered	**46.36%**	37.75%	36.31%	30.68%
coordinate_structure_constraint_object_extraction_filtered	62.38%	65.12%	**65.86%**	63.22%
determiner_noun_agreement_1_filtered	97.31%	**97.74%**	52.85%	52.85%
determiner_noun_agreement_2_filtered	96.99%	**97.10%**	85.61%	82.81%
determiner_noun_agreement_irregular_1_filtered	**83.85%**	78.12%	72.25%	70.78%
determiner_noun_agreement_irregular_2_filtered	**90.00%**	87.56%	84.15%	76.59%
determiner_noun_agreement_with_adj_2_filtered	**92.24%**	90.75%	79.81%	76.94%
determiner_noun_agreement_with_adj_irregular_1_filtered	**82.45%**	77.30%	73.96%	71.17%
determiner_noun_agreement_with_adj_irregular_2_filtered	**82.38%**	78.93%	72.26%	69.88%
determiner_noun_agreement_with_adjective_1_filtered	**94.96%**	91.00%	51.77%	51.55%
distractor_agreement_relational_noun_filtered	**86.29%**	45.05%	68.40%	57.11%
distractor_agreement_relative_clause_filtered	**58.09%**	43.17%	50.98%	57.41%
drop_argument_filtered	75.76%	**75.98%**	60.87%	62.07%
ellipsis_n_bar_1_filtered	51.50%	**56.36%**	54.36%	53.87%
ellipsis_n_bar_2_filtered	58.09%	**63.29%**	43.36%	49.64%
existential_there_object_raising_filtered	**81.65%**	72.66%	79.80%	68.10%
existential_there_quantifiers_1_filtered	**99.46%**	97.42%	96.77%	93.76%
existential_there_quantifiers_2_filtered	28.21%	33.92%	38.42%	**43.69%**
existential_there_subject_raising_filtered	83.98%	82.90%	**84.31%**	80.84%
expletive_it_object_raising_filtered	70.09%	**73.12%**	72.46%	70.22%
inchoative_filtered	**55.79%**	52.28%	44.91%	46.67%
intransitive_filtered	**68.32%**	67.17%	46.31%	50.58%
irregular_past_participle_adjectives_filtered	**94.80%**	88.14%	72.84%	63.58%
irregular_past_participle_verbs_filtered	81.53%	81.10%	**85.14%**	77.39%
irregular_plural_subject_verb_agreement_1_filtered	**83.33%**	76.62%	82.21%	72.14%
irregular_plural_subject_verb_agreement_2_filtered	**89.46%**	87.33%	88.00%	83.86%
left_branch_island_echo_question_filtered	65.15%	61.67%	63.15%	**70.86%**
left_branch_island_simple_question_filtered	**60.15%**	46.79%	57.83%	50.26%
matrix_question_npi_licensor_present_filtered	15.82%	12.38%	17.98%	**31.75%**
npi_present_1_filtered	**50.39%**	40.59%	46.75%	48.51%
npi_present_2_filtered	49.89%	**50.33%**	45.62%	48.69%
only_npi_licensor_present_filtered	**98.07%**	48.64%	76.87%	92.06%
only_npi_scope_filtered	50.90%	44.92%	61.05%	**80.53%**
passive_1_filtered	89.17%	**90.60%**	87.74%	86.79%
passive_2_filtered	88.15%	**89.37%**	83.61%	81.28%
principle_A_c_command_filtered	55.07%	**59.51%**	51.48%	59.41%
principle_A_case_1_filtered	**100.00%**	**100.00%**	**100.00%**	99.89%
principle_A_case_2_filtered	91.58%	**92.57%**	88.20%	78.80%
principle_A_domain_1_filtered	96.39%	98.25%	**100.00%**	**100.00%**
principle_A_domain_2_filtered	53.55%	50.71%	**63.61%**	51.80%
principle_A_domain_3_filtered	50.90%	50.90%	**61.00%**	55.58%
principle_A_reconstruction_filtered	41.88%	34.64%	**53.67%**	47.67%
regular_plural_subject_verb_agreement_1_filtered	**93.48%**	90.45%	88.76%	80.11%
regular_plural_subject_verb_agreement_2_filtered	**90.37%**	85.19%	82.65%	77.67%
sentential_negation_npi_licensor_present_filtered	96.19%	96.74%	**99.35%**	96.52%
sentential_negation_npi_scope_filtered	21.70%	23.08%	33.30%	**40.76%**
sentential_subject_island_filtered	40.89%	39.33%	**58.17%**	57.54%
superlative_quantifiers_1_filtered	66.70%	66.80%	**70.99%**	54.14%
superlative_quantifiers_2_filtered	76.37%	**83.77%**	69.98%	61.16%
tough_vs_raising_1_filtered	**36.50%**	28.80%	23.73%	29.32%
tough_vs_raising_2_filtered	81.41%	**82.93%**	80.76%	78.37%
transitive_filtered	**80.07%**	74.77%	70.85%	66.94%
wh_island_filtered	61.77%	**63.54%**	61.04%	38.75%
wh_questions_object_gap_filtered	78.70%	75.20%	**80.33%**	76.37%
wh_questions_subject_gap_filtered	92.32%	**92.54%**	92.43%	90.31%
wh_questions_subject_gap_long_distance_filtered	91.60%	93.35%	93.58%	**94.87%**
wh_vs_that_no_gap_filtered	95.82%	95.93%	**96.17%**	94.54%
wh_vs_that_no_gap_long_distance_filtered	94.86%	**97.37%**	96.57%	94.74%
wh_vs_that_with_gap_filtered	**27.20%**	26.01%	5.55%	7.07%
wh_vs_that_with_gap_long_distance_filtered	**7.03%**	4.18%	3.41%	4.62%
supplement_hypernym	51.19%	**51.90%**	51.07%	51.19%
supplement_qa_congruence_easy	48.44%	54.69%	56.25%	**57.81%**
supplement_qa_congruence_tricky	26.67%	**39.39%**	25.45%	25.45%
supplement_subject_aux_inversion	78.54%	77.22%	**86.11%**	79.75%
supplement_turn_taking	56.79%	**58.21%**	**58.21%**	56.43%

Exploring Curriculum Learning for Vision-Language Tasks: A Study on Small-Scale Multimodal Training

Rohan Saha[1], Abrar Fahim[1], Alona Fyshe[1,2], Alex Murphy[1]
[1]Department of Computing Science, [2]Department of Psychology
University of Alberta
{rsaha, afahim2, alona, amurphy3}@ualberta.ca

Abstract

For specialized domains, there is often not a wealth of data with which to train large machine learning models. In such limited data / compute settings, various methods exist aiming to *do more with less*, such as finetuning from a pretrained model, modulating difficulty levels as data are presented to a model (curriculum learning), and considering the role of model type / size. Approaches to efficient *machine* learning also take inspiration from *human* learning by considering use cases where machine learning systems have access to approximately the same number of words experienced by a 13 year old child (100M words). We investigate the role of 3 primary variables in a limited data regime as part of the multimodal track of the BabyLM challenge. We contrast: (i) curriculum learning, (ii), pretraining (with text-only data), (iii) model type. We modulate these variables and assess them on two types of tasks: (a) multimodal (text+image), and (b) unimodal (text-only) tasks. We find that curriculum learning benefits multimodal evaluations over non-curriclum learning models, particularly when combining text-only pretraining. On text-only tasks, curriculum learning appears to help models with smaller trainable parameter counts. We suggest possible reasons based on architectural differences and training designs as to why one might observe such results.

1 Introduction

Recent vision-language models (VLMs) have achieved superior performance on numerous benchmark datasets (such as the Llama[1] and Gemini models[2]), and continue advancing rapidly as models are scaled up. The number of parameters of such models is often on the order of billions. These models require multiple days of compute, and hundreds of GPUs (e.g., Radford et al. (2021)), resulting in massive energy consumption (Luccioni et al., 2024). Furthermore, to train such large models, we require massive amounts of pretraining data. For example, 70M image-text pairs were used to train the Flava foundation model (Singh et al., 2022). Pretraining VLMs on such large scale data is often infeasible for independent researchers and university research labs with limited compute.

In contrast to *machine* learning, *human* learning is much more efficient, a finding which has led researchers to consider which methods might promote more *human-like* learning in artificial neural networks. This was originally argued for in early work on curriculum learning (Bengio et al., 2009), citing the fact that humans do not learn from randomly sampled data, but benefit from learning over structured chunks, typically increasing in difficulty (a curriculum).

To this end, we explore the application of curriculum learning to VLMs with limited input data as part of the BabyLM challenge (Choshen et al., 2024). For the multimodal track, which contains a dataset of image-caption pairs, we take inspiration from phase-based curriculum methodology used in Ayyubi et al. (2023). We use Part-of-Speech (PoS) linguistic features from the captions to categorize samples into different phases, to generate a learning curriculum. However, instead of training the model only one phase at a time (as used in Ayyubi et al. (2023)), we train the model on the current and previous phases such that the pool of data which can be sampled increases at each phase.

From our experiments, we observe that:

- In a limited data setting, curriculum learning can improve the performance of VLMs on certain multimodal and text-only evaluation benchmarks.
- Pretraining VLMs on developmentally plausible text-only data prior to adapting to multimodal data may help improve performance on some evaluation tasks, but not others.

[1]https://llama.meta.com/
[2]https://deepmind.google/technologies/gemini/

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 65–81
November 15-16, 2024 ©2024 Association for Computational Linguistics

2 Background

2.1 Curriculum Learning

Curriculum Learning (CL) takes inspiration from the learning process in humans by presenting data to a machine learning model in an easy-to-difficulty manner (Elman, 1993; Bengio et al., 2009). CL consists of two parts: (1) a scoring function to rank data samples based on difficulty, and (2) a pacing function, which controls the distribution of data samples presented to the model. In the standard CL implementation, the pacing function introduces can be samples in ascending order of difficulty (or decreasing difficulty in the case of anti-curriculum learning (Hacohen and Weinshall, 2019; Wu et al., 2021)).

While extensive research has shown that in certain cases, curriculum learning can provide performance gains in vision (Hacohen and Weinshall, 2019; Wang et al., 2019b; Soviany, 2020) and Natural Language Processing (NLP) tasks (Nagatsuka et al., 2021; Maharana and Bansal, 2022; Sun et al., 2023), in other cases, the benefit is unclear (Campos, 2021; Martinez et al., 2023; Chobey et al., 2023; Edman and Bylinina, 2023a). Importantly, with the prevalence of vision-language models, it is crucial to understand how the application of CL modulates VLMs to work in the domain of limited data and compute.

2.2 Curriculum Learning for Vision Language Models

Some previous work has applied CL to multimodal models where the data modality consists of images and texts. Srinivasan et al. (2023) showed that CL applied to a transformer model helps improve performance on zero-shot image and text retrieval tasks over a baseline CLIP model (Radford et al., 2021). CL has also shown benefit in other multimodal domains, such as medical report generation (Liu et al., 2023), image-captioning (Ayyubi et al., 2023), and visual question answering (Li et al., 2020). However, these works either rely on non vision-transformer based image encoders (such as an R-CNN), or conduct evaluation on a small set of evaluation tasks. It is also unclear whether: (i) training VLMs on image-caption data improves model performance on text-only benchmarks; (ii) how CL affects downstream performance in models with additional text pretraining compared to randomly initialized models.

In this work, we present a study where we apply CL to VLMs trained on *small data*. We hope to provide the research community with a better understanding of the effects of CL on popular VLMs such as the Generative Image Transformer (GIT) (Wang et al., 2022) and Flamingo (Alayrac et al., 2022) models. Furthermore, we also explore the effect of CL on downstream model performance on various zero-shot multimodal and text-based benchmarks.

3 Methods

3.1 Data

We use the dataset provided as part of the BabyLM multimodal track (Choshen et al., 2024). The data consist of 100M words in total: 50M words from varied text corpora (described in Choshen et al. (2024)) and the other 50M words are text captions taken from the Conceptual Captions (Sharma et al., 2018) and Localized Narratives (Pont-Tuset et al., 2020) image-caption datasets. In total, the multimodal data consists of $\sim$ 2.9M image-caption pairs.

One of the key experimental variables we examine is the impact of text pretraining. For multimodal models, we compare the performance of models trained on image-caption data (consisting of 50M words), starting either from a randomly initialized model or from a model pretrained on the text-only corpora mentioned above (50M words). Model variants not pretrained on the text-only corpora only use the words in the captions of the associated training images (i.e., models are trained on only 50M words and the corresponding images).

3.2 Models

We train two VLMs: (1) GIT Wang et al. (2022) and (2) Flamingo (Alayrac et al., 2022). We chose these models because they were selected as reference baselines provided by the BabyLM challenge (Choshen et al., 2024). Both GIT and $Flamingo$ models consist of vision encoders to encode image inputs, and text decoders to generate free-form text.

We use the default configurations for the GIT[3] and $Flamingo$[4] models provided in the BabyLM challenge to compare the performance of our models to the baselines reported by the challenge. Following the default configurations, we use pretrained vision encoders[5] for both the GIT and

[3] https://huggingface.co/babylm/git-2024

[4] https://huggingface.co/babylm/flamingo-2024

[5] https://huggingface.co/facebook/dino-vitb16

$Flamingo$ models. Furthermore, according to default model configurations, we update all model parameters for the GIT model, but for $Flamingo$, we keep the vision encoder frozen, and update all other parameters. GIT has a total of 198 million parameters (198 million trainable parameters), and $Flamingo$ has 255 million total parameters (169 million trainable parameters because of the frozen vision encoder).

Tokenizer: Pretrained tokenizers are trained on data that exceed the limit imposed by the challenge. Thus, we train a new *WordPiece* tokenizer (using a `bert-base-uncased` model configuration) from scratch on the text-only and caption data (100M words total). We use the same tokenizer for both GIT and $Flamingo$ to avoid confounding model performance differences with the tokenizer choice.

3.3 Curriculum Framework

We discuss the respective implementations of the scoring and pacing functions for the curriculum learning framework below.

Scoring function: A scoring function assigns a *difficulty score* $k \in \mathbb{R}$ to each sample in the dataset, where a sample x_i is easier than a sample x_{i+1}, if $k_{x_i} < k_{x_{i+1}}$.

Previous works have used a variety of scoring functions to measure sample difficulty, such as the loss scoring function in image classification (Hacohen and Weinshall, 2019) and text classification settings (Xu et al., 2020; Maharana and Bansal, 2022). Relatedly, in sample-efficient pretraining of language models, average sentence rarity (Borazjanizadeh, 2023), sentence length (DeBenedetto, 2023) or other combinations of individual text statistics (Edman and Bylinina, 2023b) have been used to rank data samples (for a comprehensive survey, see Soviany et al. (2021)). More recently, in multimodal settings, cross-modal similarity (Zhang et al., 2022) has been used to rank examples to improve model performance in image-captioning tasks. All in all, it must be noted that determining the difficulty of image-caption pairs is non-trivial and an active research problem.

For our experiments, we explored the applicability of linguistic information such as Part-of-Speech (PoS) tags to determine difficulty of samples. We took inspiration from the scoring function used by Ayyubi et al. (2023), where a PoS tagger was used to count the number of nouns in the caption, as an indirect measure of the number of concepts present in the image. The number of concepts, in turn, determined the difficulty of the image-caption pair.

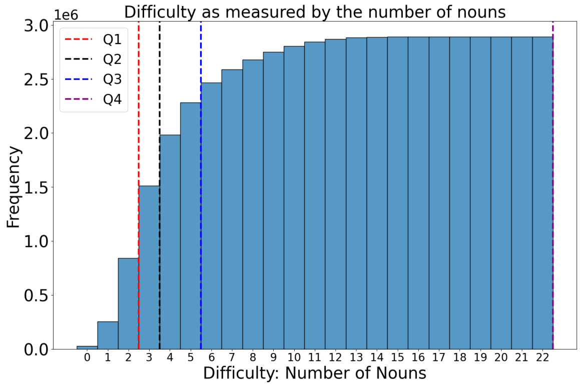


Figure 1: Cumulative distribution of scores for all the image-caption pairs. The dashed vertical lines determine each of the four quartiles, where each quartile contains the samples that belong to a specific curriculum phase.

As the BabyLM challenge has limits on the number of words that can be used to train systems, we trained our own PoS tagger to tag the image captions. To train the tagger, we first created a training dataset by annotating the provided text-only and caption data, with POS symbols[6], using an off-the-shelf PoS tagger from NLTK [7]. Then we used this newly created annotated training dataset to train a custom PoS tagger on the permissible limited text words. We implemented the PoS tagger using a token classification model using $BERT_{BASE}$ as the backbone model architecture. We trained the tagger for 5 epochs[8], using a batch size of 512 and half-precision (FP16) training.

Distribution of difficulty scores: We show the cumulative distribution of the scores assigned by the PoS scoring function in Figure 1. For images having multiple captions, we consider the maximum value of the difficulty (maximum number of nouns) amongst all the captions for that image. We use maximum difficulty to account for the most complex interpretation of the image and avoid underestimation of the difficulty value.

Ordering: In our experiments, we order the samples in ascending order of difficulty, to explore the

[6]These are non-word elements such as NN for noun, or JJ for determiner

[7]https://www.nltk.org/api/nltk.tag.pos_tag.html

[8]We observed that 5 epochs were sufficient to achieve ~97.42% accuracy on a 10% held out validation dataset. We then trained the tagger on all the data (train+validation).

performance improvement of unimodal and multimodal models when they are trained in a manner similar to how humans acquire novel information. Although previous work has shown that a descending ordering of difficulty can be beneficial for model performance for certain tasks (e.g., Maharana and Bansal (2022)), we leave this for future research given limited compute.

Pacing function: A pacing function controls the rate at which samples of different training curriculum phases are presented to the model. Multiple different pacing strategies exist, such as fixed exponential pacing, step pacing for image classification (Hacohen and Weinshall, 2019), competence function (Platanios et al., 2019) for machine translation, to name a few.

For our experiments, we design a simple pacing function inspired by the phase-level pacing function (Ayyubi et al., 2023) and competence-based pacing function (Platanios et al., 2019). We use the quartiles from the cumulative distribution of the sample difficulty scores (Figure 1), giving us four *blocks* of difficulty levels. For simplicity, we also train our model in four phases, where in each training phase p, we train the model on samples that have difficulty levels in the p^{th} quartile. For example, in Figure 1, the first phase (p_1) contains samples with difficulty level $k \leq 2$, the second phase contains samples with difficulty level $k \leq 3$, the third phase contains samples with $k \leq 5$, while the fourth phase contains all the samples in the dataset. For each training phase, we randomly sample training batches from the set of data available up to the corresponding training quartile. It must be noted with each new block, the number of available data points increases, which has an effect during training, where earlier epochs are faster (because of fewer samples) compared to later epochs.

This approach contrasts the phase-level curriculum learning introduced by Ayyubi et al. (2023), where the model is trained only on samples from a specific block, which may cause the model to focus more on samples in that specific block, while not retaining previously learned information from earlier phases. Furthermore, our pacing function has the added advantage of not requiring extensive hyperparameter tuning, such as the exponential pacing function used by Hacohen and Weinshall (2019), and is thus suitable for scenarios with limited computational resources.

3.4 Models Variants

For both GIT and $Flamingo$, we train four model variants, two of which are baseline models and two are trained using curriculum learning. In each pair, we train one model only on the image-caption data starting from random initialization (except the vision encoder which is pretrained), while we first pretrain the other variant on the text-only corpus, before training on image-caption data.

Baselines: For the first baseline variant, we train the model on the image-caption dataset (50M words) using standard i.i.d training. We refer to this variant with **C** (denoting that the model is trained on the image-caption data only) for both $GIT_{Baseline}$ and $Flamingo_{Baseline}$. For the second baseline variant, we first train the model on the text-only dataset (containing 50M words) using standard i.i.d training. We then continue the training procedure on image-caption dataset (containing another 50M words) using standard i.i.d training. We refer to this variant as **T+C**, for both $GIT_{Baseline}$ and $Flamingo_{Baseline}$.

Our choice to also train the **T+C** model variant stems from previous work showing that exposing the model to developmentally plausible data, such as child-directed speech, before exposing it to complex data, can benefit model performance (Huebner et al., 2021). Thus, we explore the difference in model performance, when we first train the model on the text-only dataset, before continuing the training procedure on the image-caption data.

Curriculum models: For curriculum variants, we use CL on the image-caption pairs because we hypothesize that applying CL on multimodal data will improve model performance. We refer to these variants trained only on the image-caption pairs as **C** under GIT_{CL} and $Flamingo_{CL}$. We also train **T+C** variants of CL models, where we first pretrain the model on the text-only dataset using standard i.i.d training, and then use curriculum learning to continue the training procedure on the image-caption pairs.

To summarize, we trained four variants for each model, two of which were trained using standard training (no curriculum), and the other two were trained using curriculum learning. For GIT and $Flamingo$ baseline variants, we train the model on the image-caption only (**C**) data, and both text + image-caption (**T+C**) data. Similarly, for the curriculum variants, we train each model on, image-caption data only (**C**) data, and both text + image (**T+C**) data.

4 Training and Evaluation Details

Training Details: For the curriculum variants, we train the model for two epochs per each *difficulty phase* (of which there are four). We used a learning rate of $1e^{-5}$, maximum token length of 50, and 32 samples per batch [9], and Adam optimizer[10] (Kingma and Ba, 2017).

When training the **T+C** variants of our baseline and curriculum models, we first trained the model on the text-only dataset for twenty epochs (instead of eight epochs for image-caption data) and use the same hyperparameter values. We used an NVIDIA A5000 GPU with 24GB vRAM, with half-precision (FP16) to train the models. We provide the total time required to train each model variant in Appendix A. For all experiments, we set the random seed to 0 to remove variation in the results due to different random sampling and initialization. We also hold out 5% of the full image-caption dataset to validate the model. We show the validation loss curves in Appendix B.

Evaluation: To evaluate the performance of our models, we use the evaluation pipeline provided by challenge (Gao et al., 2023; Choshen et al., 2024). We report the performance of all the variants of the GIT and $Flamingo$ models on the multimodal, and text-based evaluation tasks.

4.1 Multimodal evaluation datasets

Winoground : The Winoground dataset (Thrush et al., 2022) evaluates a model's ability to perform visio-linguistic compositional reasoning. Specifically, given two image-caption pairs, the goal is to match the image to the corresponding caption, where both captions contain an identical set of words, but in a different order (e.g. *It's a fire truck* vs *it's a truck fire*). The dataset consists of 400 examples with 800 unique images and captions. To assess model performance, we use the unpaired text-score metric as provided in the BabyLM evaluation pipeline.

[9]We use a batch size of 32 when training on the image-caption data, but we use a value of 256 when pretraining the model (T+C variant) on the text-only dataset as memory requirements are lower.

[10]We use default hyperparameters for Adam: β_1 = 0.9, β_2 = 0.999, eps=$1e^{-08}$, weight_decay=0.

VQAv2: The VQAv2 dataset (Goyal et al., 2017) is a large-scale visual question answering dataset. It contains open-ended questions about images, requiring models to understand the visual content and generate appropriate answers. We use accuracy as the choice of metric as reported in the BabyLM evaluation pipeline. For this task the model has to select the best answer for a given image and question, in the presence of 7 distractors.

DevBench: The DevBench dataset (Tan et al., 2024) is a multimodal benchmark for developmental evaluation that evaluates how closely a model's outputs align with human responses. It includes tasks such as object recognition, action recognition, and visual question answering, using data from both children and adults. The BabyLM pipeline uses three tasks from the DevBench dataset: (1) The (Lexical) Visual Vocabulary (lex-viz_vocab) task involves selecting the correct image from several image options based on a given word. (2) The (Grammatical) Test of Reception of Grammar (gram-trog) task involves choosing the correct image based on a sentence, testing grammatical understanding using distractor images that correspond to sentences with different word orderings (e.g. "a white cat sitting on a brown couch" vs. "a brown cat sitting on a white couch"). Finally, (3) the (Semantic) THINGS Similarity (sem-things) task uses Representational Similarity Analysis (RSA) to compare the model's image similarity judgments with human responses.

4.2 Text-only evaluation datasets

BLIMP (and BLIMP Supplement): The `BLIMP` dataset (Warstadt et al., 2020) is a benchmark for evaluating syntactic and semantic knowledge in language models. It consists of sentences with systematic variations in syntax and semantics. The `BLIMP Supplement` extends the original dataset with additional challenging examples.

(Super)GLUE: The `(Super)Glue` benchmark (Wang et al., 2018, 2019a) is a collection of diverse natural language understanding tasks designed to evaluate a model's ability to perform well across multiple domains and evaluates generalized linguistic ability. The BabyLM challenge includes tasks, `COLA`, `SST2`, `MRPC`, `QQP`, `MNLI`, `MNLI-MM`, `QNLI`, `RTE` from the `GLUE` benchmark, and the tasks `BoolQ`, `RTE` and `WSC` from `SuperGLUE` benchmark. To fine tune all our model variants, we use a train batch size of 128, validation batch size of 16,

Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
	C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
Winoground	54.02	55.50	51.34	55.23	50.00	51.21	51.21	50.80
VQAv2	41.22	41.72	42.84	43.98	41.99	43.00	35.93	40.85

Table 1: Results for baseline and curriculum models on the Winoground and VQAv2 evaluation datasets. **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
	C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
lex-viz_vocab	72.27	75.63	78.15	73.11	66.39	52.94	58.82	54.62
gram-trog	32.89	38.16	32.29	39.47	34.21	34.21	34.21	35.53
sem-things	33.39	25.79	22.83	32.08	46.46	47.99	50.21	51.66
Avg: $devbench_{acc}$	46.18	46.52	44.63	48.22	49.02	45.05	47.75	47.27

Table 2: Accuracy results for baseline and curriculum models on the DevBench dataset. RSA scores are used for sem-things **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants..

Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
	C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
lex-viz_vocab	68.25	68.59	70.19	70.66	64.47	57.63	63.08	57.46
gram-trog	44.46	46.51	44.77	45.79	43.59	42.77	42.54	43.29
sem-things	33.39	25.79	22.83	32.08	46.46	47.99	50.21	51.66
Avg: $devbench_{hs}$	48.70	46.96	45.93	49.51	51.51	49.46	51.94	50.80

Table 3: Human similarity scores for baseline and curriculum models on the DevBench dataset. RSA scores are used for sem-things. **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

learning rate of $5e^{-5}$, early stopping patience of 3, maximum sequence length of 50, and maximum number of epochs=10. We used default values for all other hyperparameters provided in the BabyLM evaluation pipeline.

EWOK: The EWOK dataset (Ivanova et al., 2024) is a zero-shot dataset for evaluating compositional generalization in language models. It consists of sentences with compositional structures that require models to generalize to unseen combinations of words and syntactic patterns.

5 Results

As unimodal and multimodal tasks are qualitatively different, we analyze the three experimental variables of interest (curriculum, pretraining & model type) in the context of each task type. Namely, we report the results for all variants of GIT and $Flamingo$ models across two main task types that differ with respect to their data inputs: (i) multimodal (image+captions), and (i) unimodal (text-only).

5.1 Multimodal (image+captions)

We show the multimodal evaluations results in Table 1 for Winoground and VQAv2, and in Tables 2 (accuracy) and 3 (human similarity) for DevBench.

5.1.1 Curriculum Learning

The GIT_{CL} model performs better than $GIT_{Baseline}$ on VQAv2 and DevBench datasets,

Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
	C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BliMP Supp	44.29	52.89	48.61	51.24	44.24	52.59	45.71	53.28
BLiMP filtered	57.85	62.90	61.34	64.05	57.03	59.82	55.64	60.13
(Super)$GLUE_{avg}$	59.96	61.12	59.79	61.46	59.82	62.79	60.53	64.29
$EWOK_{avg}$	50.62	51.55	49.82	50.98	50.03	50.67	50.16	50.71

Table 4: Average results for the text-only evaluation datasets.**C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

with and without pretraining on separate text data. This is not the case for `Winoground`, which we note has quite unique properties, such as specifically probing model representations for compositionality (see Section 4.1).

We find that $Flamingo_{CL}$ only performs better than its associated baseline ($Flamingo_{Baseline}$) on the `DevBench` dataset when using accuracy, and when evaluating using human response scores. This result indicates that curriculum training may benefit multimodal model performance when evaluated on benchmark datasets that focus on developmentally plausible evaluation of language models.

5.1.2 Text Pretraining

Compared to training on just image-caption data, pretraining with the text-only data (variant **T+C**) produces higher scores across both $GIT_{Baseline}$ and GIT_{CL} models on `Winoground` and `DevBench`, while the results are more mixed for $Flamingo$ models. However, in $Flamingo_{CL}$ on the `VQAv2` dataset, we see the largest gain in performance due to text pretraining (from 35.93 to 40.85, a gain of $\sim 5\%$ in Table 1). On the `DevBench` evaluation for GIT_{CL}, we also see the 2nd largest gain in performance due to text pretraining (from 44.63 to 48.22 for accuracy, and from 45.93 to 49.51 when using reference human similarity scores; a gain of $\sim 4\%$). Interestingly, the highest result of all models on the `Winoground` dataset are the GIT models with text pretraining, suggesting that text-only pretraining is a big contributor to the properties of the `Winoground` evaluation benchmark (compositionality). However, one must be cautious about generalizing this finding as the performance increase could simply result from the model being trained on more data.

As we only use a single seed to report these results, we wanted to confirm that our observation is not simply due to random chance. Thus, we conduct more experiments where we train all GIT variants using two more seeds, and observe a similar pattern in our findings (text pretraining aids model performance). We provide these results in Appendix C.

5.1.3 Model Type

The two models differ in their application of attention mechanism and model size, measured by the number of trainable parameters (See Section 3.2). $Flamingo$ has a frozen image encoder (unlike GIT) and cross-attention is applied prior to each LM block in the Transformer stack (which internally contains the standard self-attention mechanism). In contrast, GIT uses a projection module to bring image embeddings into the same space as the text embeddings and applies successive self-attention on these vectors. We see multiple variants of GIT outperform $Flamingo$ (especially for `Winoground`, `VQAv2`, and `lex-viz_vocab`, `gram-trog` subsets for `DevBench`). In the multimodal evaluation context, we believe this could be due to the ability for GIT to update the parameters of its vision encoder, perhaps additionally by making use of the fact that image tokens can self-attend to one another (unlike the cross-attention in $Flamingo$, which does not have this property).

5.2 Unimodal (text-only)

We summarize the results for the unimodal (text-only) evaluation in Table 4. This table contains summary results for the three text-only evaluation benchmarks (see Section 4.2). Table 9 contains detailed results on the `(Super)GLUE` and `EWOK` benchmarks. We also provide a detailed breakdown of model performance for each text-based task in Appendix D.

5.2.1 Curriculum Learning

Closely related to the observations for multimodal benchmarks, we see that curriculum learning variants outperform corresponding baselines variants on the unimodal (text-only) benchmarks. Although both GIT_{CL} and $Flamingo_{CL}$ outperformed their corresponding baselines (Tables 4 and 9), the effect was greater in $Flamingo_{CL}$.

5.2.2 Text Pretraining

We outline the averaged results in Table 4 and show that for both $Flamingo$ and GIT, text pretraining leads to a gain in performance. In fact, all **T+C** variants (curriculum and baseline) for both models showe better performance compared to **C** variants. Coupled with curriculum learning, we observe performance benefits on all text-based evaluation datasets. These results suggest that text pretraining conveys a clear advantage for multimodal models when they are evaluated on certain text-based benchmarks.

5.2.3 Model Type

Unlike the multimodal results, considering the average results in Table 4, there was no consistent pattern where one model type outperformed the other. For example, on `(Super)GLUE`, both baseline and CL **T+C** variants of $Flamingo$ outperformed respective GIT variants. However, this was not the case for `BLIMP filtered`, where we observed the opposite pattern - all variants of GIT outperformed all variants of $Flamingo$. Such a result could result from the fact that both GIT and $Flamingo$ become more similar in their architecture in the text-only evaluation setting. This can stem relaxed requirement to incorporate image information, making both models resemble standard autoregressive Transformer decoders (the trainable parameter count changes in this context because GIT's vision encoder was trainable in the multimodal case, while $Flamingo$'s was frozen). This results in the trainable parameter count for GIT being 198M and 169M for $Flamingo$ (Section 3.2).

5.3 Brief Summary of Results

For the multimodal evaluation, we observe that text pretraining before image-caption training boosts model performance compared to no text pretraing. However, these observations must be cautiously generalized across model types; text pretraining largely conveys a benefit in all GIT models, but this benefit is inconsistent for $Flamingo$. For instance, the $Flamingo_{CL}$ variant benefits from additional text-only pretraining over just image-caption training (for `VQAv2`, `gram-trog`, and `sem-things`), but this effect is unclear for the $Flamingo_{Baseline}$. For GIT model variants, curriculum learning (combined with pretraining) resulted in the best overall model scores on `VQAv2` and `DevBench` (considering average scores in Tables 2 and 3).

For the text-only evaluation, removing the image component from both the GIT and $Flamingo$ models effectively reduces them to text-only transformer architectures with differing number of parameters. This likely explains why the models show similar performance across tasks despite their original multimodal design. Nonetheless, we see that in Table 4, the $Flamingo_{CL}$ **T+C** variant can be more suited to learning representations leading to better scores across the `SuperGLUE` benchmark, and `BLiMP supplement` dataset. But on `BLiMP filtered` (and less pronounced for `EWOK`), the **T+C** variant of GIT_{CL} outperforms the **T+C** variant of $Flamingo_{CL}$.

Conclusion

In this study, we explore the application of a curriculum learning (CL) approach to training vision-language models (VLMs) in a limited data setting. We use a custom trained Part-of-Speech (PoS) tagger to determine the complexity of image-caption pairs. We train two variants for each of the GIT and $Flamingo$ models using curriculum learning and compare their performance against variants trained using standard i.i.d training. We find that while CL training shows potential, its benefits are not universally applicable across all GIT and $Flamingo$ variants. However, for certain model configurations, CL enhances performance on a range of downstream, multimodal and text-based tasks (zero-shot and finetuning). Importantly, pretraining VLMs on developmentally plausible text data prior to multimodal training can contribute to performance gains. Nonetheless, generalizing this result requires careful consideration, as factors such as model architecture, training data composition, and the nature of evaluation tasks can significantly affect model performance.

Code and Data Availability

We release our code, model predictions, and model checkpoints.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning. *Preprint*, arXiv:2204.14198.

Hammad A Ayyubi, Rahul Lokesh, Alireza Zareian, Bo Wu, and Shih-Fu Chang. 2023. Learning from children: Improving image-caption pretraining via curriculum. *arXiv preprint arXiv:2305.17540*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Nasim Borazjanizadeh. 2023. Optimizing GPT-2 pretraining on BabyLM corpus with difficulty-based sentence reordering. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 356–365, Singapore. Association for Computational Linguistics.

Daniel Campos. 2021. Curriculum learning for language modeling. *arXiv preprint arXiv:2108.02170*.

Aryaman Chobey, Oliver Smith, Anzi Wang, and Grusha Prasad. 2023. Can training neural language models on a curriculum with developmentally plausible data improve alignment with human reading behavior? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 98–111, Singapore. Association for Computational Linguistics.

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. *Computing Research Repository*, arXiv:2404.06214.

Justin DeBenedetto. 2023. Byte-ranked curriculum learning for babylm strict-small shared task 2023. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 198–206.

Lukas Edman and Lisa Bylinina. 2023a. Too much information: Keeping training simple for babylms. *arXiv preprint arXiv:2311.01955*.

Lukas Edman and Lisa Bylinina. 2023b. Too much information: Keeping training simple for BabyLMs. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 89–97, Singapore. Association for Computational Linguistics.

Jeffrey L. Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *Preprint*, arXiv:1612.00837.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *International conference on machine learning*, pages 2535–2544. PMLR.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

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*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Qing Li, Siyuan Huang, Yining Hong, and Song-Chun Zhu. 2020. A competence-aware curriculum for visual concepts learning via question answering. In *European Conference on Computer Vision*, pages 141–157. Springer.

Fenglin Liu, Shen Ge, Yuexian Zou, and Xian Wu. 2023. Competence-based multimodal curriculum learning for medical report generation. *Preprint*, arXiv:2206.14579.

Sasha Luccioni, Yacine Jernite, and Emma Strubell. 2024. Power hungry processing: Watts driving the cost of ai deployment? In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and*

Transparency, FAccT '24, page 85–99, New York, NY, USA. Association for Computing Machinery.

Adyasha Maharana and Mohit Bansal. 2022. On curriculum learning for commonsense reasoning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 983–992, Seattle, United States. Association for Computational Linguistics.

Richard Diehl Martinez, Hope McGovern, Zebulon Goriely, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – Curriculum Learning for Infant-inspired Model Building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 84–99, Singapore. Association for Computational Linguistics.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2021. Pre-training a BERT with curriculum learning by increasing block-size of input text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 989–996, Held Online. INCOMA Ltd.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. 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 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *ECCV*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*.

Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. 2022. Flava: A foundational language and vision alignment model. *Preprint*, arXiv:2112.04482.

Petru Soviany. 2020. Curriculum Learning with Diversity for Supervised Computer Vision Tasks. *Preprint*, arXiv:2009.10625.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2021. Curriculum Learning: A Survey. *arXiv preprint*.

Tejas Srinivasan, Xiang Ren, and Jesse Thomason. 2023. Curriculum learning for data-efficient vision-language alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5619–5624.

Shichao Sun, Ruifeng Yuan, Jianfei He, Ziqiang Cao, Wenjie Li, and Xiaohua Jia. 2023. Data selection curriculum for abstractive text summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7990–7995.

Alvin Wei Ming Tan, Sunny Yu, Bria Long, Wanjing Anya Ma, Tonya Murray, Rebecca D. Silverman, Jason D. Yeatman, and Michael C. Frank. 2024. Devbench: A multimodal developmental benchmark for language learning. *Preprint*, arXiv:2406.10215.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language. *Preprint*, arXiv:2205.14100.

Yiru Wang, Weihao Gan, Jie Yang, Wei Wu, and Junjie Yan. 2019b. Dynamic curriculum learning for imbalanced data classification. *Preprint*, arXiv:1901.06783.

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.

Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. 2021. When do curricula work? *Preprint*, arXiv:2012.03107.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Hongkuan Zhang, Saku Sugawara, Akiko Aizawa, Lei Zhou, Ryohei Sasano, and Koichi Takeda. 2022. Cross-modal similarity-based curriculum learning for image captioning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7599–7606, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Comparison of Training Times

We show the comparison of the training times for different baseline and curriculum variants in Table 5.

Model	Variant	Hours
$GIT_{Baseline}$	C	~ 80
	T+C	~ 109
GIT_{CL}	C	~ 50
	T+C	~ 79
$Flamingo_{Baseline}$	C	~ 79
	T+C	~ 105
$Flamingo_{CL}$	C	~ 46
	T+C	~ 72

Table 5: Comparison of training times amongst all model variants. These training times include validation loss calculation after every epoch. The pretraining on the text-only dataset (for the T+C variants) accounted for about 29 hours for the GIT model and around 26 hours for the $Flamingo$ model. Curriculum models take fewer hours to train because of the dynamic nature of the training data size that grows during training.

B Validation loss curves

We show the validation loss curves on a held out 5% of the image-caption data in Figure 2.

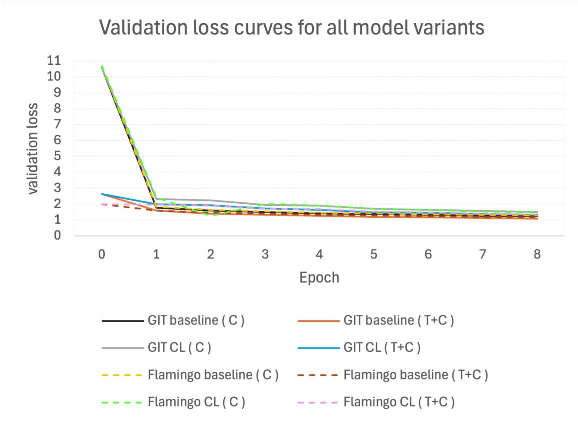


Figure 2: Validation loss curves for all the model variants. GIT variants are shown in solid lines and $Flamingo$ variants are shown in dashed lines. The x-axis denotes the epochs, and the value at the 0th epoch denotes the validation loss of the model before being trained on the image-caption pairs (i.e., before training on the first epoch). For the **T+C** variants, since the model is pretrained on the text-only dataset before being trained on the image-caption pairs, the loss starts at a lower value compared to the model variants on image-caption data only (**C**) that were randomly initialized.

C GIT model multimodal results across 3 seeds

We show the multimodal evaluation results for the different GIT model variants in Tables 6 for Winoground and VQAv2, 7 for accuracy on DevBench, and 8 human similarity scores on DevBench.

Tasks	$GIT_{Baseline}$		GIT_{CL}	
	C	**T+C**	**C**	**T+C**
Winoground	54.02	53.71	51.52	54.65
VQAv2	38.80	41.90	42.28	42.60

Table 6: Results for GIT baseline and GIT curriculum models on the multimodal evaluation datasets averaged across three seeds. **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

Tasks	$GIT_{Baseline}$		GIT_{CL}	
	C	**T+C**	**C**	**T+C**
lex-viz_vocab	72.93	72.55	75.91	71.71
gram-trog	38.16	36.84	32.26	41.67
sem-things	30.88	25.61	17.34	30.78
$Average_{acc}$	47.32	45.00	41.84	48.05

Table 7: Accuracy results for GIT baseline and GIT curriculum models on the devbench datasets averaged across three seeds. **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

Tasks	$GIT_{Baseline}$		GIT_{CL}	
	C	**T+C**	**C**	**T+C**
lex-viz_vocab	68.64	68.07	68.65	68.71
gram-trog	44.90	44.61	43.72	45.71
sem-things	30.88	25.61	17.34	30.78
$Average_{hs}$	48.14	46.10	43.24	48.40

Table 8: Human similarity results for GIT baseline and GIT curriculum models on the devbench datasets averaged across three seeds. **C**: Model trained on image-caption pairs only (50M words), **T+C**: the model is first trained on the text-only dataset (20 epochs) and then trained on image-caption pairs (50M+50M=100M words). Green cells: winning variants over corresponding baseline variants.

D Evaluation results on `(Super)GLUE`, `EWOK`, and `BLiMP`

We show the results for all models and corresponding variants on each individual subtask for the text-only evaluation tasks in Tables 9 for (Super)GLUE and EWOK, 10 for BLiMP Supplement, and 11 , 12, 13, 14 for BLiMP.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
$SuperGLUE_{ft}$	boolq	64.04	65.2	64.04	70.21	67.77	66.91	68.07	66.54
	cola (mcc)	6.68	6.68	0.0	6.68	0.0	17.7	0.0	31.75
	mnli	68.7	69.74	69.34	69.93	66.24	70.03	67.07	70.37
	mnli-mm	69.43	70.22	69.26	70.77	66.9	70.2	66.35	71.42
	mrpc (f1)	82.12	82.13	81.23	81.35	81.05	82.51	79.87	82.39
	multirc	55.57	57.43	57.55	56.97	60.81	53.55	58.21	56.23
	qnli	63.14	64.42	67.5	65.59	65.81	68.92	67.86	69.91
	qqp (f1)	80.92	81.7	80.12	81.53	79.83	82.05	79.91	81.88
	rte	46.04	48.92	46.04	46.04	46.04	52.52	56.12	46.04
	sst2	84.40	87.39	84.17	88.53	85.09	87.84	83.94	88.30
	wsc	38.46	38.46	38.46	38.46	38.46	38.46	38.46	42.31
EWOK	agent prop	50.05	50.14	50.09	49.59	49.46	50.32	49.91	49.68
	mat-dynam	51.56	52.21	51.30	50.65	49.48	52.21	50.52	54.42
	mat-prop	50.59	52.35	47.06	49.41	46.47	53.53	51.76	51.18
	phy-dynam	49.17	55.83	48.33	58.33	54.17	48.33	50.0	51.67
	phy-inter	49.64	50.0	50.18	50.18	50.18	49.1	48.74	49.1
	phy-relation	50.24	49.88	50.61	49.51	52.57	50.12	49.27	50.86
	quant-prop	51.91	50.96	49.68	50.96	49.36	53.5	50.64	50.0
	social-interac	50.34	50.34	50.34	49.66	49.32	49.32	49.66	50.0
	social-prop	50.3	50.91	50.91	50.0	50.0	49.09	50.61	50.0
	social-relation	50.32	51.42	49.94	50.0	49.29	50.0	50.45	50.06
	spatial-relation	52.65	53.06	49.59	52.45	50.0	51.84	50.20	50.82

Table 9: Breakdown of model performance on each subtask for the (Super)Glue and EWOK datasets. Cells highlighted in Green denote winning variants compared to corresponding baseline variants.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BLiMP Supplement	hypernym	47.86	48.81	49.76	48.93	49.17	48.93	48.1	51.19
	qa_congruence_easy	29.69	51.56	35.94	50.0	32.81	51.56	37.5	53.12
	qa_congruence_tricky	27.88	24.24	27.27	20.0	20.0	30.91	27.27	28.48
	subject_aux_inversion	66.02	83.76	80.06	83.68	68.53	81.54	71.4	82.91
	turn_taking	50.0	56.07	50.0	53.57	50.71	50.0	44.29	50.71
	Average	44.29	52.89	48.61	51.24	44.24	52.59	45.71	53.28

Table 10: Breakdown of model performance on each subtask for the BLiMP Supplement dataset. Cells highlighted in green denote winning variants compared to corresponding baseline variants.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BLiMP	determiner_noun _agreement_with_adj _irregular_1	64.62	74.51	71.87	76.32	50.56	62.53	49.86	67.69
	principle_A_domain_3	51.75	51.97	48.67	51.22	48.46	48.57	49.31	45.59
	sentential_negation _npi_scope	47.65	61.31	57.52	55.57	56.83	54.54	55.57	50.86
	complex_NP_island	41.13	51.89	41.61	54.37	58.87	43.5	62.17	41.13
	irregular_plural _subject_verb _agreement_1	55.35	64.68	63.06	64.18	49.5	57.71	51.87	60.45
	distractor_agreement _relational_noun	41.62	46.7	47.21	51.27	52.03	46.83	49.37	47.46
	matrix_question _npi_licensor_present	3.98	44.78	4.2	33.05	84.5	59.85	35.84	39.5
	passive_2	70.65	70.32	72.54	72.2	70.32	70.1	72.09	64.12
	adjunct_island	78.45	64.12	48.38	66.38	55.6	59.81	63.25	56.03
	wh_vs_that_with_gap	16.1	26.55	8.05	25.9	12.19	14.47	35.8	17.74
	irregular_past _participle_adjectives	59.63	66.6	79.19	63.68	46.51	48.8	45.37	67.01
	drop_argument	71.96	74.02	73.8	76.41	70.87	70.0	70.11	68.91
	principle_A_domain_2	49.62	57.7	57.16	59.02	46.34	50.93	50.82	56.28
	anaphor_gender _agreement	45.21	46.04	36.77	47.79	74.46	47.79	42.33	39.55
	wh_questions_subject _gap_long_distance	93.0	85.53	97.9	89.5	81.68	88.8	61.38	89.96
	only_npi_licensor _present	61.68	74.72	93.99	52.72	72.22	92.52	97.05	58.28
	intransitive	54.84	60.02	53.57	61.98	57.49	59.1	57.6	60.14
	ellipsis_n_bar_1	43.64	49.88	52.37	59.6	38.4	61.97	51.0	52.12
	regular_plural_subject _verb_agreement_1	44.16	58.54	53.15	58.76	49.44	55.84	45.96	61.12
	principle_A_domain_1	84.57	93.0	96.83	91.79	57.99	93.0	93.22	80.42
	irregular_past _participle_verbs	63.8	65.39	58.6	59.45	61.04	66.56	49.26	68.05
	sentential_subject _island	54.63	62.12	67.33	56.71	53.69	51.93	49.84	63.89

Table 11: BLIMP - individual task results. Cells highlighted in Green denote winning variants compared to corresponding baseline variants.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BLiMP	wh_vs_that_with_gap _long_distance	13.52	10.0	5.49	10.22	15.6	8.46	40.77	12.2
	principle_A_recons-truction	54.6	50.36	53.05	35.26	56.05	53.26	50.47	55.43
	regular_plural_subject_verb _agreement_2	55.03	66.88	64.44	68.25	48.99	51.43	51.22	61.9
	ellipsis_n_bar_2	29.59	51.93	31.76	53.26	37.92	45.41	33.57	55.68
	determiner_noun _agreement_with _adj_irregular_2	65.36	75.71	70.12	77.5	60.0	65.12	57.26	68.33
	passive_1	78.1	71.55	80.48	76.19	70.36	75.83	77.02	71.9
	irregular_plural _subject_verb _agree-ment_2	59.64	68.61	71.86	67.94	48.88	60.09	55.83	69.06
	existential_there _sub-ject_raising	54.11	75.65	56.06	77.81	59.74	67.42	55.3	71.21
	left_branch_island _echo_question	52.69	18.69	61.14	18.27	22.39	23.34	6.65	33.37
	expletive_it_object _raising	63.9	63.77	62.32	62.45	63.37	64.16	61.92	63.77
	coordinate_structure _constraint_object _extraction	36.14	33.4	51.74	53.74	40.99	50.26	46.36	61.54
	causative	58.07	67.48	56.48	70.17	52.57	60.15	50.12	59.78
	npi_present_2	38.4	61.38	45.19	58.64	46.28	61.6	26.15	44.64

Table 12: BLIMP - individual task results continued. Cells highlighted in Green denote winning variants compared to corresponding baseline variants.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BLiMP	animate_subject_trans	46.05	44.53	22.64	38.68	30.55	49.84	64.46	66.31
	transitive	69.93	73.04	71.08	75.23	52.65	63.59	60.25	58.99
	determiner_noun _agreement_with_adj_2	65.99	78.53	65.57	81.62	50.05	60.04	56.11	70.24
	determiner_noun _agreement_irregular_2	75.12	81.34	72.2	84.88	63.17	73.78	61.71	77.56
	left_branch_island _simple_question	46.37	36.8	62.78	35.44	39.54	45.53	33.96	37.64
	wh_vs_that_no_gap	85.13	91.17	94.19	94.89	90.36	93.26	64.0	93.26
	tough_vs_raising_2	67.72	69.24	74.57	72.5	51.74	63.7	56.85	72.07
	principle_A_case_1	99.78	100.0	99.78	100.0	93.31	98.79	98.25	98.03
	wh_questions_subject_gap	81.51	85.41	91.43	88.86	82.63	89.2	72.16	87.53
	only_npi_scope	35.72	50.3	69.3	46.12	79.81	61.05	75.03	39.67
	distractor_agreement _relative_clause	43.51	46.73	40.07	44.78	54.31	48.91	53.16	48.56
	existential_there _quantifiers_2	58.29	17.34	38.31	30.63	19.87	34.03	21.08	18.33
	determiner_noun _agreement_1	74.27	81.92	71.69	84.39	56.51	70.72	58.56	75.03
	superlative_quantifiers_1	61.08	71.71	48.52	85.39	51.17	39.43	57.3	37.59
	determiner_noun _agreement_with_adjective_1	64.84	80.49	69.77	81.89	56.81	63.88	57.56	71.28
	sentential_negation _npi_licensor_present	90.64	99.35	99.56	92.49	91.95	99.56	72.91	98.91
	wh_questions_object_gap	55.65	49.71	73.69	57.97	73.11	64.96	72.53	60.3
	determiner_noun _agreement_2	69.92	80.88	71.21	82.92	52.52	66.38	57.14	75.94
	existential_there _quantifiers_1	78.06	92.15	77.96	94.52	75.48	66.77	74.73	68.6
	inchoative	43.04	50.53	40.12	52.16	43.63	49.01	44.91	50.76
	coordinate_structure _constraint_complex_left _branch	40.07	30.13	55.08	27.37	35.76	38.41	33.11	30.13
	superlative_quantifiers_2	86.51	75.56	88.03	79.11	78.19	48.68	76.27	46.96
	npi_present_1	40.48	52.59	53.14	57.43	48.4	57.98	50.72	57.87
	wh_island	17.71	27.92	32.08	51.88	61.25	18.12	48.75	40.42
	existential_there_object _raising	70.44	66.13	67.73	60.96	68.23	70.94	66.26	67.98

Table 13: BLIMP - individual task results continued. Cells highlighted in Green denote winning variants compared to corresponding baseline variants.

	Tasks	$GIT_{Baseline}$		GIT_{CL}		$Flamingo_{Baseline}$		$Flamingo_{CL}$	
		C	**T+C**	**C**	**T+C**	**C**	**T+C**	**C**	**T+C**
BLiMP	wh_vs_that_no_gap_long _distance	86.4	94.4	94.97	96.57	89.6	94.29	61.37	93.37
	principle_A_c_command	69.13	71.88	66.07	75.58	57.61	75.69	66.17	78.12
	animate_subject_passive	61.45	70.28	73.85	72.18	63.13	65.14	60.67	72.51
	anaphor_number_agre-ement	73.15	80.34	62.41	86.14	71.0	72.82	49.41	74.22
	determiner_noun _agree-ment_irregular_1	64.61	70.63	67.25	75.18	59.47	62.56	54.63	73.57
	tough_vs_raising_1	33.12	49.89	28.69	46.62	51.9	46.41	47.36	39.45
	principle_A_case_2	62.84	77.27	72.35	79.23	54.97	62.95	48.96	62.62

Table 14: BLIMP - individual task results continued. Cells highlighted in Green denote winning variants compared to corresponding baseline variants.

BABYHGRN: Exploring RNNs for Sample-Efficient Training of Language Models

Patrick Haller **Jonas Golde** **Alan Akbik**

Humboldt-Universität zu Berlin
{patrick.haller.1, jonas.max.golde, alan.akbik}@hu-berlin.de

Abstract

This paper explores the potential of recurrent neural networks (RNNs) and other subquadratic architectures as competitive alternatives to transformer-based models in low-resource language modeling scenarios. We utilize HGRN2 (Qin et al., 2024), a recently proposed RNN-based architecture, and comparatively evaluate its effectiveness against transformer-based baselines and other subquadratic architectures (LSTM, xLSTM, Mamba). Our experimental results show that BABYHGRN, our HGRN2 language model, outperforms transformer-based models in both the 10M and 100M word tracks of the challenge, as measured by their performance on the BLiMP, EWoK, GLUE and BEAR benchmarks. Further, we show the positive impact of knowledge distillation. Our findings challenge the prevailing focus on transformer architectures and indicate the viability of RNN-based models, particularly in resource-constrained environments.

1 Introduction

In recent years, natural language processing (NLP) has been revolutionized by transformer-based language models (LMs), like BERT (Devlin et al., 2019) or GPT (Brown et al., 2020) and their derivatives, achieving state-of-the-art results (Touvron et al., 2023; Abdin et al., 2024) across a wide range of tasks such as machine translation, question answering, and text generation. However, despite their dominance, transformers come with notable limitations: they require extensive training data (Hoffmann et al., 2022) and enormous computational resources, which pose challenges for their use in resource-constrained environments.

These limitations led to an increasing interest in more sample-efficient alternatives and approaches with lower computational requirements (Wang et al., 2020b). The shared task of the BabyLM Challenge (Warstadt et al., 2023a) systematically explores this trend by training LMs on datasets of limited size (10M words in the "strict-small" and 100M words in the "strict" setup). The resulting models are then evaluated on linguistic and general language understanding tasks.

While most participants in BabyLM Challenge focus on adapting transformers to low-resource settings, we propose revisiting recurrent neural networks (RNNs). Once foundational to sequence modeling tasks (Lample et al., 2016; Howard and Ruder, 2018), RNNs have been largely overshadowed by transformers due to their sequential nature which does not easily allow for parallelization.

Potential of RNN-architectures. In this paper, we investigate whether the inductive biases of RNN architectures, such as their sequential processing and memory states, provide advantages in data-constrained settings. This question is especially relevant given that state-of-the-art transformer models depend on quadratic self-attention, which requires calculating the inner product between all tokens. In particular, we investigate the potential of the HGRN2 (Qin et al., 2024), a novel subquadratic RNN-based architecture based on hierarchical gating. We train our model using knowledge distillation (Hinton et al., 2015) and evaluate our approach, BABYHGRN, against state-of-the-art transformer models and other efficient RNN architectures (e.g. xLSTM (Beck et al., 2024) or Mamba (Gu and Dao, 2024)). Our experiments demonstrate that our resulting model yields better performance compared to both transformer-based and other RNN-based architectures.

We summarize our contributions as follows:

1. We conduct an exploratory evaluation of transformer-based and other RNN-based architectures (HGRN2, LSTM, xLSTM, Mamba), contributing to the ongoing research on sample-efficient language modeling.

2. We present a comprehensive evaluation of our

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 82–94
November 15-16, 2024 ©2024 Association for Computational Linguistics

Dataset	Count	Ratio (%)
Pile-CC	4,900,155	49.00
OpenWebText2	3,078,791	30.79
FreeLaw	946,382	9.46
USPTO Backgrounds	261,159	2.61
Wikipedia (en)	187,094	1.87
PubMed Central	142,698	1.43
PubMed Abstracts	118,427	1.18
Others	365,188	3.65
Total	9,999,894	

Dataset	Count	Ratio (%)
Pile-CC	49,214,555	49.21
OpenWebText2	30,344,790	30.34
FreeLaw	9,471,436	9.47
USPTO Backgrounds	2,519,390	2.52
Wikipedia (en)	1,855,709	1.86
PubMed Central	1,449,273	1.45
PubMed Abstracts	1,175,838	1.18
Others	3,968,870	3.97
Total	99,999,861	

Table 1: Composition of the **10M** (left table) and **100M** (right table) word datasets (word counts and ratio per domain) we created from the PILE to train BABYHGRN.

proposed HGRN2 language model BABYHGRN. We show the impact of knowledge distillation and the choice of dataset.

3. We release all code, datasets, and experimental setups to the research community to facilitate reproducibility and further research[1].

Our results show that BABYHGRN outperforms transformer-based baselines on both tracks of the BabyLM challenge.

2 BABYHGRN

We utilize HGRN2 as our backbone architecture with a hidden size of 2048 and 18 layers, resulting in a total parameter count of 330M. We train our model either with (*1*) the default dataset of the BabyLM Challenge or (*2*) a sub-sampled split of ThePile (Gao et al., 2020). Further, we employ knowledge distillation training using a teacher-student setup. In the following, we will discuss the details of our design choices.

2.1 Training Dataset

We curate our own training datasets for the `strict` and `strict-small` tracks by sub-sampling the Pile dataset (see Table 1). The Pile consists of 22 smaller datasets that cover a variety of domains, including books, web pages, scientific literature, and programming code. The main motivation behind choosing the Pile dataset is its diverse composition, which may offer several advantages for language model training. Approximately 14% of the original BabyLM dataset consists of child-related text (e.g., the Children's Book Test (Hill et al., 2016), Children's Stories Text Corpus[2], and CHILDES project (Macwhinney, 2000)), which may limit its generalizability across diverse domains. In contrast, the broader scope of the Pile dataset could improve resilience in zero-shot tasks and potentially enhance adaptability for fine-tuning on specific areas of interest.

We create the splits by randomly sampling from each chosen subset until we reached the pre-defined thresholds. We depict details on our selected subsets and corresponding word counts in Table 1.

To minimize computational overhead, we concatenate all samples and segment them into uniform chunks of 512 tokens. Subsequently, each input sample is tokenized using Byte-Pair Encoding (BPE), employing a vocabulary size of 16,000 tokens. We chose the *BabyLlama* tokenizer provided with the baseline models by the organizers[3].

2.2 Training Objectives

We use standard next-token prediction as the language modeling task and employ token-level cross-entropy loss for training our models. For a sequence of tokens $x = (x_1, ..., x_N)$, the loss is calculated as:

$$\mathcal{L}(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log P(x_i|x_1, ..., x_{i-1}; \theta)$$

where θ represents the model parameters and $P(x_i \mid x_1, ..., x_{i-1}; \theta)$ is the probability the model assigns to the i-th token given all previous tokens.

We further improve our model through knowledge distillation (Bucila et al., 2006; Hinton et al.,

[1] https://github.com/HallerPatrick/BabyLM-2024

[2] https://www.kaggle.com/datasets/edenbd/children-stories-text-corpus

[3] https://huggingface.co/babylm/babyllama-100m-2024

2015), where we train a second HGRN2 model (student) using predictions from our initially trained model (teacher). While knowledge distillation traditionally transfers knowledge from larger to smaller models, using same-sized teacher and student models has proven effective in recent work - notably in the previous BabyLM Challenge where an ensemble of teachers was used for knowledge transfer (Timiryasov and Tastet, 2023).

The training process for the student model incorporates an additional loss term based on soft labels produced by the teacher model. The total loss function for the student model can be expressed as:

$$\mathcal{L}_{\text{total}} = (1 - \alpha)\mathcal{L}_{\text{CE}} + \alpha\mathcal{L}_{\text{KD}}$$

where $\mathcal{L}_{\text{CE}}$ is the standard cross-entropy loss for the student model, $\mathcal{L}_{\text{KD}}$ is the knowledge distillation loss, and α is a hyperparameter that balances the two loss terms.

In our implementation, the knowledge distillation loss $\mathcal{L}_{\text{KD}}$ is calculated using the Kullback-Leibler divergence between the probability distributions of the teacher and student models:

$$\mathcal{L}_{\text{KD}} = \text{KL}(\sigma(z_t)||\sigma(z_s))$$

where z_t and z_s are the output logits of the teacher and student model respectively. And $\sigma(z)$ is the softmax function applied to the logits z.

2.3 Training Details

For fine-tuning on the (Super)Glue tasks, we follow the provided hyperparameters by the shared task organizer (see Appendix A). Except for the *WSC* tasks, which had unusually low scores. We used a maximum of 20 epochs, a patience of 6 epochs and a learning rate of 1×10^{-5} for our final submission models.

Software. For training our model we use the `Pytorch` (Ansel et al., 2024) library. Relevant metrics are logged with `Weights and Biases` (Biewald, 2020). We use HuggingFace `datasets` (Lhoest et al., 2021) library for dataset loading and subsampling. All relevant models were either directly imported with the `transformers` (Wolf et al., 2020) library or implemented as a custom model. For the HGRN2 model we used the FLA (Yang and Zhang, 2024) library.

Hardware. All models were trained with the `torch.distributed` package in data-parallel mode. Models were trained on 4 RTX A6000 49GB graphics cards on one node.

3 Empirical Evaluation

In Section 3.1, we shortly present the evaluation benchmarks of the BabyLM Challenge and the BEAR knowledge probe. In Sections 3.2 to 3.4, we evaluate BabyHGRN compared with other efficient RNN architectures, its training dynamics, and the influence of different datasets. Finally, in Section 3.5, we evaluate BabyHGRN using knowledge distillation.

3.1 Evaluation Datasets

The BabyLM challenge covers three benchmarks: BLiMP (Warstadt et al., 2023b), EWoK (Ivanova et al., 2024), and parts of GLUE (Wang et al., 2019) and SuperGLUE (Wang et al., 2020a), respectively. These benchmarks are designed to assess language model performance such as grammatical knowledge or complex reasoning tasks. Additionally, we include the BEAR probe (Wiland et al., 2024) to evaluate factual knowledge capabilities.

BLiMP (Benchmark of Linguistic Minimal Pairs) is an English zero-shot benchmark evaluating the grammatical knowledge of language models. It has 67 sub-tasks, each focusing on a specific syntactic or semantic phenomenon. Specifically, the dataset contains pairs of sentences and the model is tasked to differentiate which of the sentences is grammatically correct. Further, we consider the hidden task *"BLiMP Supplement"* of the 2023 BabyLM Challenge (Warstadt et al., 2023a).

EWoK (Elements of World Knowledge) evaluates basic world knowledge in language models. This cognition-inspired approach tests whether language models can identify plausible contexts given different fillers. EWoK was introduced as the hidden task for the 2024 BabyLM Challenge.

GLUE (General Language Understanding Evaluation) is a multi-task benchmark evaluating natural language understanding systems. It contains nine tasks such as sentiment analysis, question answering, or textual entailment. As models began to surpass human performance on several GLUE tasks, SuperGLUE was introduced as an extension, including more challenging tasks.

BEAR (Wiland et al., 2024) tests relational knowledge in language models using 7,731 instances over 60 relations. BEAR compares the models' log-likelihood for different factual statements of which only one is true. We leverage the implementation by Ploner et al. (2024) to conduct the BEAR probing experiments.

Model	#Params	Epoch	BLiMP	BLiMP-Supp.	EWoK	Macro-Avg.
Transformer	360M	4	62.64	54.86	50.48	55.99
LSTM	300M	5	62.27	51.63	50.48	54.79
xLSTM	340M	3	51.20	48.77	49.89	50.02
Mamba	350M	2	64.44	55.39	50.39	56.74
HGRN2	360M	4	67.05	55.69	49.88	**57.54**

Table 2: Results from training on the 10M word corpus, comparing various RNN architectures to a Transformer-based model (LLaMA architecture). Each model was trained for 5 epochs, with evaluations after each epoch, and the best-performing model was selected.

Hyperparameter	Value
Epochs	3
Batch Size	64
Learning Rates	{1e-3, 1e-4, 1e-5, 1e-6}
Optimizer	Adam
Sequence Length	512
Max Grad Norm	1.0
LR Scheduler	Linear

Table 3: Pretraining hyperparameters used for all models and experiments.

3.2 Experiment 1: RNN Architecture Selection

Our first experiment compares the HGRN architecture with other RNN-based and transformer architectures. Specifically, we compare HGRN2, the vanilla LSTM, xLSTM, Mamba, and a Transformer baseline.

Experimental setup. We select configurations such that all architectures have a similar parameter count of 300 to 360 million. We use the configurations as as originally proposed for xLSTM, Mamba, and HGRN2. For the decoder-only transformer, we use the LLaMA (Touvron et al., 2023) model and follow the Pythia (Biderman et al., 2023) 410M model configuration with 22 hidden layers. For the vanilla LSTM, we set the hidden size to 4096 with two layers to match the parameter count of the other architectures. We refer to Appendix B for a detailed overview of all configurations.

For each architecture, we perform learning rate selection for all considered architectures by executing a grid search over commonly used learning rates ({1e-3, 1e-4, 1e-5, 1e-6}). We train each model for 5 epochs on the `strict-small` dataset of the BabyLM challenge. Further, we do not employ any knowledge distillation and train all LMs using the next-token prediction objective. We report results on the zero-shot benchmarks of BabyLM, namely BLiMP and EWoK, together with their best hyperparameter configuration.

Results. Table 2 shows the number of parameters of each considered architecture and the results achieved during the exploration phase on the zero-shot benchmarks[4]. We find that the HGRN2 exhibits the best performance, closely followed by Mamba. Both outperform the transformer model, suggesting that these architectures offer advantages in low-resource scenarios. The standard LSTM, serving as a baseline for classical RNN architectures, performs worse than the transformer but better than the xLSTM[5]. Further, we observe that all architectures perform best using a learning rate of $1e^{-3}$.

3.3 Experiment 2: Learning Dynamics of HGRN2

To better understand the learning dynamics of the selected HGRN2 architecture, we investigated how its zero-shot performance on the BabyLM benchmark changes over the epochs during training.

Experimental setup. We re-use the best performing hyperparameters from Section 3.2. After each epoch, we evaluate on BLiMP, BLiMP Supp. and EWoK.

Results. The results of this experiment are illustrated in Figure 1. Our analysis reveals early peaks in performance on BLiMP and EWoK and a later peak on BLiMP Supplement. This finding indicates that HRGN2 initially captures certain linguistic patterns from the limited training data, although the

[4] We report the complete results of the parameter sweep in Appendix C.

[5] We note considerable training instabilities during pretraining of the xLSTM model, likely due to discrepancies in model architecture or training setup from the original implementation, which may have impacted performance in our low-resource setting.

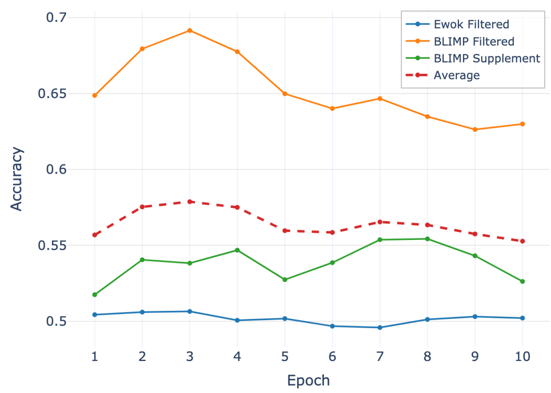


Figure 1: Performance evaluation of epochs of pretraining, with the macro average at epoch 3 being the highest.

gains over random baseline are modest. Further iterations yield only incremental improvements, which may point to constraints in the model's ability to leverage the available data fully.

3.4 Experiment 3: Impact of Training Dataset

In this experiment, we evaluate the impact of the choice of training data. We compare models trained over the default BabyLM dataset to models trained using our custom dataset derived from the Pile (see section 2.1).

Experimental Setup. We re-use our chosen hyperparameter configuration for the HGRN2 architecture from Section 3.3 and train two models on (*1*) our derived Pile subset and (*2*) on the default BabyLM dataset. We train models for 5 epochs, evaluate after each epoch, and report results of the best performing model. In this experiment we include both the 10M and 100M word datasets for a full comparison.

Results. Table 4 summarizes the performance across all benchmarks. For the 10M word track, the HGRN2 model trained on our derived dataset shows modest gains on BLiMP, EWoK, and BEAR, but underperforms on the BLiMP-Supplemental subset (↓3.47 pp). This suggests that at smaller data scales, our dataset may lack certain syntactic structures present in the original BabyLM dataset. Furthermore, given the limited dataset size in the 10M word track, these numbers may lack statistical significance.

In contrast, the 100M word track demonstrates consistently stronger performance across all metrics, with particularly notable improvements on

Dataset	BLiMP	BLiMP-Supp.	EWoK	BEAR
BabyLM - 10M	67.05	55.69	49.88	5.29
Ours - 10M	67.49	52.22	50.62	5.36
BabyLM - 100M	69.44	55.56	50.31	6.17
Ours - 100M	**72.89**	**57.43**	50.61	**7.38**

Table 4: Zero-shot evaluation results comparing HGRN2 models trained on the BabyLM dataset versus our proposed Pile subset. Both models were trained with a learning rate of 1×10^{-3}. All metrics are reported as percentages.

BLiMP (↑3.45 pp) and BEAR (↑1.21 pp). Indicating that our dataset selection strategy enhances the model's ability to acquire both syntactic and factual knowledge when given sufficient training data.

3.5 Experiment 4: BabyHGRN With Knowledge Distillation

Based on the exploratory experiments of the previous subsections, we selected the HGRN2 model trained on our proposed dataset for the BabyLM challenge. We furthermore apply knowledge distillation as outlined in Section 2.2 to our final model. We refer to this model as BabyHGRN.

In this section, we evaluate BabyHGRN using knowledge distillation learning and compare it with two baselines (BabyLlama and LTG-BERT) and a BabyHGRN version using only the *n*ext-*t*oken *p*rediction objective. We denote this ablation model as BabyHGRN$_{ntp}$.

Hyperparameters. We increase the model size in accordance with scaling laws for language models (Kaplan et al., 2020) from 360M to 1.0B. We reduce the learning rate from 1×10^{-3} to 4×10^{-4} accordingly, following the configuration found in Sections 3.2 and 3.3. Empirical work (Kaplan et al., 2020; Hoffmann et al., 2022) suggests that lower learning rates in larger models help mitigate instabilities during training, promoting smoother convergence and more efficient use of computational resources.

3.5.1 Results

Table 5 and Table 6 summarize our experimental results for the 10M and 100M word tracks, respectively.

HGRN2 outperforms baselines. Most importantly, we find that our HGRN2 models show competitive performance across both the 10M and 100M word tracks of the BabyLM challenge. On the 10M words track, BabyHGRN achieves an overall macro average of 63.3% (↑2.5 pp vs. BabyL-

	BLiMP	BLiMP-Supp.	EWoK	SuperGLUE	Average	BEAR
BabyLlama	69.8	59.5	50.7	63.3	60.8	5.4
LTG-BERT	60.6	60.8	48.9	60.3	57.7	5.7
BabyHGRN$_{ntp}$ *(ours)*	69.4	55.6	50.7	63.0	59.7	5.6
BabyHGRN *(ours)*	72.1	58.6	51.3	65.8	**63.3**	7.5

Table 5: Evaluation results for the **10M words** track ("strict-small"). The BabyLM score is computed as a macro average over four datasets (BLiMP, BLiMP Supp., EWoK and SuperGLUE) but note that the macro average may not be a representative overall score for each model, since the datasets are of widely varying size (e.g. the BLiMP supplements is only 7% in size compared to the BLiMP). We additionally include the BEAR score for comparison and evaluation of factual knowledge.

	BLiMP	BLiMP-Supp.	EWoK	SuperGLUE	Average	BEAR
BabyLlama	73.1	60.6	52.1	69.0	63.7	8.5
LTG-BERT	69.2	66.5	51.9	68.4	64.0	8.2
BabyHGRN$_{ntp}$ *(ours)*	74.5	59.1	52.88	69.1	63.9	13.5
BabyHGRN *(ours)*	77.5	58.5	51.6	70.7	**64.9**	13.6

Table 6: Evaluation results for the **100M words** track ("strict"). The BabyLM score is computed as a macro average over four datasets (BLiMP, BLiMP Supp., EWoK and SuperGLUE). We additionally include the BEAR score for comparison and evaluation of factual knowledge.

lama). As Table 5 shows, BabyHGRN particularly outperforms the baselines on the BLiMP (↑2.4 pp vs. BabyLlama) and SuperGLUE (↑2.5 pp vs. BabyLlama) tasks, and significantly improves the BEAR score (↑1.8 pp vs. LTG-BERT).

On the 100M words track (refer to Table 6), BabyHGRN outperforms the baselines with a marco average of 64.9% (↑0.9 pp vs. LTG-BERT), though the improvement is not as pronounced as in the more data-constrained 10M scenario. Here, BabyHGRN improves in particular the BLiMP (↑4.4 pp vs. LTG-BERT) and SuperGLUE (↑1.7 pp vs. BabyLlama) tasks, but falls short on BLiMP-Supplement (↓7.4 pp vs. LTG-BERT)[6].

Knowledge distillation is helpful. We also note that our knowledge distillation approach significantly improves performance of BabyHGRN, compared to the distillation-free approach BabyHGRN$_{ntp}$. As Tables 5 and 6 show, BabyHGRN outperforms both, BabyLlama and LGT-BERT, baselines. Further, we observe BabyHGRN outperforms BabyHGRN$_{ntp}$ by 5.3 pp on average in the data-constrained 10M setting, confirming the usefulness of distillation losses in such settings.

[6]Detailed results for BLiMP, BLiMP-Supplement, EWoK and (Super)Glue are provided in Appendix D.

BabyHGRN is better at learning factual knowledge. While the accuracy on BEAR is relatively low across all settings (compared to state-of-the-art models such as LLaMA-3 with 68.6), we observe that BabyHGRN strongly outperforms transformer-based baselines in data-restricted settings. For instance, BEAR shows a pronounced difference between BabyHGRN and BabyHGRN$_{ntp}$ on the 10M track, and a large difference between the HGRN models and the baselines on the 100M track. We primarily attribute this improvement to the use of our custom dataset.

4 Related Work

In recent years, there has been a resurgence of interest in recurrent neural network (RNN) architectures for sequence modeling, particularly in the context of large language models (LLMs). This renewed focus has led to the development of several RNN-based architectures that aim to combine the efficiency of recurrent models with the expressiveness of more complex architectures like transformers.

HGRN and HGRN2 The Hierarchically Gated Recurrent Neural Network (HGRN) (Qin et al., 2023) introduces a novel gating mechanism that allows for more effective modeling of long-term dependencies. The key innovation of HGRN is its hierarchical structure, in which forget gates have monotonically increasing lower bound values from bottom layers to upper layers. This design enables lower layers to model short-term dependencies while upper layers capture long-term relationships in the data. HGRN achieves efficient training by reformulating its recurrent computation as a parallel scan operation to enable parallelization across sequence length while maintaining linear time complexity.

Building upon HGRN, Qin et al. (2024) introduced HGRN2 which further enhances the capabilities of gated linear RNNs. HGRN2 addresses some limitations of its predecessor by incorporating a state expansion mechanism. This innovation significantly increases the recurrent state size without introducing additional parameters, leading to improved expressiveness.

xLSTM Another recently proposed RNN-based architecture is the Extended Long Short-Term Memory (xLSTM) (Beck et al., 2024). xLSTM builds upon the classical LSTM (Hochreiter and Schmidhuber, 1997) by introducing two key modifications: exponential gating and modified memory structures. The exponential gating mechanism allows the model to revise storage decisions more effectively, addressing a key limitation of traditional LSTMs. xLSTM introduces two variants: sLSTM with a scalar memory and new memory mixing technique, and mLSTM with a matrix memory and covariance update rule, which is fully parallelizable. The xLSTM approach demonstrates strong performance across various modalities, including language, vision (Alkin et al., 2024; Chen et al., 2024), and audio (Yadav et al., 2024), while maintaining linear scaling in sequence length and efficient inference.

The **Mamba** architecture (Gu and Dao, 2024) improves on state space models (SSMs) by introducing selective state spaces. Building on structured SSMs (Gu et al., 2022), Mamba achieves linear-time sequence processing through input-dependent SSM parameters, enabling selective information propagation across sequences. This mechanism is conceptually similar to gating in classical RNNs (Hochreiter and Schmidhuber, 1997) while maintaining modern computational benefits. The architecture consists of repeated blocks that combine selective SSMs with feed-forward components, in contrast to more complex predecessors like H3 (Fu et al., 2023) and Hyena (Poli et al., 2023). Though attention-free, Mamba matches or exceeds Transformer performance (Vaswani et al., 2023) across various domains. Its recurrent computation pattern eliminates the need for attention caches during inference, leading to 5× faster inference compared to similar-sized Transformers. This combination of linear scaling and efficiency, without sacrificing model quality, makes Mamba a significant development in sequence modeling.

The development of HGRN2, xLSTM, and Mamba is part of a broader trend in revisiting and improving RNN architectures (Peng et al., 2023; Sun et al., 2023).

5 Conclusion

We presented BabyHGRN, an RNN-based language model that utilizes the HGRN2 architecture. Our experimental results on the evaluation datasets of the BabyLM Challenge and the BEAR probe indicate that BabyHRGN is competitive. Indeed, despite relatively little hyperparameter optimization, our approach significantly outperforms strong transformer-based baselines on the evaluation datasets.

Revisiting our research question posed in Section 1, we conclude that RNN-based language models are indeed competitive in low-resource language modeling scenarios. Based on these results, we believe that advanced RNN-based architectures such as HGRN and Mamba may hold promise for research in sample-efficient language modeling. Accordingly, future work could explore further optimizations of the underlying RNN architectures, investigate their performance on a broader range of tasks, and examine their scalability to larger datasets and model sizes.

Limitations

Our experiments with HGRN2 in the BabyLM Challenge demonstrate the competitiveness of RNN-based models with transformers in low-resource scenarios. However, while we find our results to be promising, it's important to acknowledge that there are several avenues for optimization that we have yet to explore:

Dataset sampling The dataset we used to train BabyHGRN was produced using a naive random sampling of the PILE dataset. More sophisticated approaches, such as importance sampling specialized for downstream tasks, would likely yield better results, especially if optimized for the tasks BabyLM evaluates on. In our work, we refrained from such "dataset engineering" and focused solely on a comparison of different RNN architectures.

Model configurations We utilized the configurations provided by the authors of HGRN2 and xLSTM. Further experimentation with different architectures and hyperparameters for the low-resource scenario could well lead to improved performance of these models.

Context length Optimizing the context length for our specific tasks and data could potentially enhance the model's capabilities. Work from previous years challenge (Edman and Bylinina, 2023; Cheng et al., 2023) suggests that a smaller context size improves performance on all benchmarks.

Knowledge distillation As previously discussed, we only implemented a basic knowledge distillation approach to train BabyHGRN. More sophisticated techniques, such as those employed by Timiryasov and Tastet (2023) could further boost performance.

Our work thus serves as a proof of concept, demonstrating that RNNs can be competitive with transformers in this domain, while leaving room for further advancements.

Acknowledgements

We thank all reviewers for their valuable comments. Alan Akbik and Patrick Haller are supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Emmy Noether grant "Eidetic Representations of Natural Language" (project number 448414230). Alan Akbik is furthermore supported under Germany's Excellence Strategy "Science of Intelligence" (EXC 2002/1, project number 390523135). Jonas Golde is supported by the Bundesministerium für Bildung und Forschung (BMBF) as part of the project "FewTuRe" (project number 01IS24020).

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Benedikt Alkin, Maximilian Beck, Korbinian Pöppel, Sepp Hochreiter, and Johannes Brandstetter. 2024. Vision-lstm: xlstm as generic vision backbone. *arXiv preprint arXiv:2406.04303*.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian

Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. 2024. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS '24, page 929–947, New York, NY, USA. Association for Computing Machinery.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2024. xlstm: Extended long short-term memory. *Preprint*, arXiv:2405.04517.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *Preprint*, arXiv:2304.01373.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

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.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. volume 2006, pages 535–541.

Tianrun Chen, Chaotao Ding, Lanyun Zhu, Tao Xu, Deyi Ji, Yan Wang, Ying Zang, and Zejian Li. 2024. xlstm-unet can be an effective 2d 3d medical image segmentation backbone with vision-lstm (vil) better than its mamba counterpart. *Preprint*, arXiv:2407.01530.

Ziling Cheng, Rahul Aralikatte, Ian Porada, Cesare Spinoso-Di Piano, and Jackie CK Cheung. 2023. McGill BabyLM shared task submission: The effects of data formatting and structural biases. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 207–220, Singapore. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Lukas Edman and Lisa Bylinina. 2023. Too much information: Keeping training simple for BabyLMs. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 89–97, Singapore. Association for Computational Linguistics.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. 2023. Hungry hungry hippos: Towards language modeling with state space models. *Preprint*, arXiv:2212.14052.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. *Preprint*, arXiv:2312.00752.

Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently modeling long sequences with structured state spaces. *Preprint*, arXiv:2111.00396.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. *Preprint*, arXiv:1511.02301.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80.

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. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

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 Akyürek, 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. *Preprint*, arXiv:2405.09605.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Brian Macwhinney. 2000. The childes project: tools for analyzing talk. *Child Language Teaching and Therapy*, 8.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Jiaju Lin, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanislaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. 2023. Rwkv: Reinventing rnns for the transformer era. *Preprint*, arXiv:2305.13048.

Max Ploner, Jacek Wiland, Sebastian Pohl, and Alan Akbik. 2024. Lm-pub-quiz: A comprehensive framework for zero-shot evaluation of relational knowledge in language models. *Preprint*, arXiv:2408.15729.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y. Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models. *Preprint*, arXiv:2302.10866.

Zhen Qin, Songlin Yang, Weixuan Sun, Xuyang Shen, Dong Li, Weigao Sun, and Yiran Zhong. 2024. Hgrn2: Gated linear rnns with state expansion. *Preprint*, arXiv:2404.07904.

Zhen Qin, Songlin Yang, and Yiran Zhong. 2023. Hierarchically gated recurrent neural network for sequence modeling. *Preprint*, arXiv:2311.04823.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models. *Preprint*, arXiv:2307.08621.

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. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Preprint*, arXiv:1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. *Preprint*, arXiv:1804.07461.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020b. Linformer: Self-attention with linear complexity. *Preprint*, arXiv:2006.04768.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell, editors. 2023a. *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Singapore.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2023b. Blimp: The benchmark of linguistic minimal pairs for english. *Preprint*, arXiv:1912.00582.

Jacek Wiland, Max Ploner, and Alan Akbik. 2024. BEAR: A unified framework for evaluating relational knowledge in causal and masked language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2393–2411, Mexico City, Mexico. 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, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Sarthak Yadav, Sergios Theodoridis, and Zheng-Hua Tan. 2024. Audio xlstms: Learning self-supervised audio representations with xlstms. *Preprint*, arXiv:2408.16568.

Songlin Yang and Yu Zhang. 2024. Fla: A triton-based library for hardware-efficient implementations of linear attention mechanism.

A Finetune Hyperparameters

Hyperparameter	Value
Initial learning rate	5e-5
Batch size	64
Maximum epochs	10
Evaluate every (epochs)	1
Patience	3

Figure 2: Default hyperparameters for fine-tuning on the (Super)Glue tasks.

B Model Configurations

Transformer	Value
Hidden Size	1024
Intermediate Size	4096
Hidden Layers	22
Attention Heads	32
LSTM	Value
Hidden Size	9120
Embedding Size	512
LSTM Layers	2
Dropout	0.1
xLSTM	Value
Embedding Size	1024
Num Blocks	48
mLSTM Heads	4
sLSTM Heads	4
sLSTM BLocks at	[3, 5, 7, 40, 42, 44]
Mamba	Value
Hidden Size	1024
Intermediate Size	2048
Hidden Layers	48
State Size	8
HGRN2 - 360M	Value
Hidden Size	1024
Layers	26
Hidden Ratio	4
Expand Ratio	128
HGRN2 - 1.2B	Value
Hidden Size	2048
Layers	18
Hidden Ratio	4
Expand Ratio	128

Table 7: Complete list of model configurations.

C Learning Rate Parameter Sweep

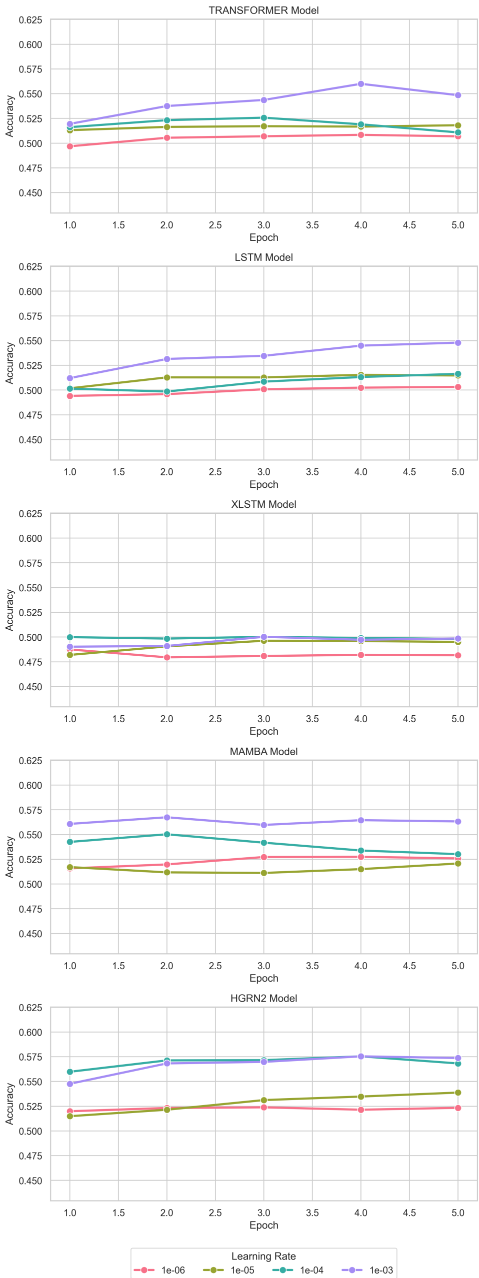


Figure 3: Evaluation results of learning rate sweep over different architectures. Scores are reported as the macro average over the three zero-shot benchmarks BLiMP, BLiMP-Supplement and EWoK.

D Final BabyLM Evaluation Scores

We provide detailed scores of all SuperGLUE, BLiMP-Supplement and EWoK tasks in Tables 8, 9 and 10. Due to the large number of subtasks in BLiMP, we will make the scores accessible though our Github repository: https://github.com/HallerPatrick/BabyLM-2024.

Model (variant)	SuperGLUE											
	BoolQ	CoLA (MCC)	MNLI	MNLI-MM	MRPC (F1)	MulitRC	QNLI	QQP (F1)	RTE	SST-2	WSC	Average
Strict-small Track (10M Words)												
BabyLlama$_{baseline}$	65.0	2.2	72.4	74.2	82.0	60.1	82.8	83.6	49.6	86.2	38.5	63.3
LTG-BERT$_{baseline}$	68.8	0.0	68.9	68.9	82.2	58.5	76.5	34.2	58.3	85.1	61.5	60.3
HGRN2	63.8	19.1	**68.7**	**68.7**	82.5	63.4	64.7	79.9	58.9	85.5	38.5	63.0
HGRN2$_{distilled}$	65.4	33.1	69.3	69.5	81.0	59.7	72.3	81.9	54.0	89.4	48.1	**65.8**
Strict-small Track (100M Words)												
BabyLlama$_{baseline}$	66.1	37.3	75.6	76.2	86.8	62.1	83.1	84.5	60.4	88.3	38.5	69.0
LTG-BERT$_{baseline}$	61.7	34.6	77.7	78.1	83.1	52.6	78.2	86.7	46.8	91.5	61.5	68.4
HGRN2	64.4	39.9	**74.3**	**74.3**	82.8	61.4	79.9	83.1	58.9	89.6	51.6	69.1
HGRN2$_{distilled}$	64.8	40.3	74.8	75.9	81.5	61.4	81.5	84.1	58.3	90.1	65.4	**70.7**
Majority Labels$_{val}$	64.0	69.9	35.7	-	68.1	57.7	50.9	62.7	53.9	51.8	61.5	57.6

Table 8: Detailed results for every task in die (Super)GLUE benchmark for the `strict` and `strict-small` track.

Model	Hypernym	QA congruence (easy)	QA congruence (tricky)	Subj.-Aux. Inversion	Turn Taking	Average
Strict-small Track (10M Words)						
BabyLlama	49.6	54.7	41.2	86.0	66.1	59.5
LTG-BERT	54.2	62.5	49.1	79.9	58.2	**60.8**
HGRN2$_{distill}$	49.8	56.2	37.6	89.6	59.6	58.6
Strict Track (100M Words)						
BabyLlama	45.6	56.2	44.8	83.9	72.5	60.6
LTG-BERT	55.0	75.0	53.3	87.5	61.4	**66.5**
HRN2$_{distill}$	48.6	64.1	35.8	84.9	59.3	58.5

Table 9: Detailed results for the BLiMP-Supplement benchmark for the `strict` and `strict-small` track.

Model	Agent Properties	Material Dynamics	Material Properties	Physical Dynamics	Physical Interactions	Physical Relations	Quantitative Properties	Social Interactions	Social Properties	Social Relations	Spatial Relations	Macroaverage
Strict-small Track (10M Words)												
BabyLlama	50.5	51.7	49.4	54.2	50.4	50.6	53.5	50.7	50.3	49.8	46.7	50.7
LTG-BERT	50.2	51.0	45.3	42.5	49.1	51.0	48.1	51.7	53.4	50.6	45.3	48.9
HGRN2$_{distll}$	50.1	50.9	50.6	55.0	50.7	50.4	51.3	54.1	51.2	50.3	49.8	**51.3**
Strict Track (100M Words)												
BabyLlama	50.1	55.5	50.0	57.5	51.4	50.5	56.7	52.7	49.7	50.0	49.0	**52.1**
LTG-BERT	50.1	55.8	50.6	58.3	48.9	50.9	53.8	51.4	50.8	53.8	49.2	51.9
HGRN2$_{distll}$	50.2	52.5	51.8	49.2	51.4	50.6	54.5	51.4	57.0	49.7	49.6	51.6

Table 10: Detailed results for the EWoK benchmark for the `strict` and `strict-small` track.

Choosy Babies Need One Coach: Inducing Mode-Seeking Behavior in Baby Llama with Reverse KL Divergence

Shaozhen Shi **Yevgen Matusevych** **Malvina Nissim**
CLCG, University of Groningen
{shaozhen.shi, yevgen.matusevych, m.nissim}@rug.nl

Abstract

This study presents our submission to the Strict-Small Track of the 2nd BabyLM Challenge. We use a teacher–student distillation setup with the Baby Llama model (Timiryasov and Tastet, 2023) as a backbone. To make the student's learning process more focused, we replace the objective function with a reverse Kullback–Leibler (KL) divergence, known to cause mode-seeking (rather than mode-averaging) behaviour in computational learners. We further experiment with having a single teacher (instead of an ensemble of two teachers) and implement additional optimization strategies to improve the distillation process. Our experiments show that under reverse KL divergence, a single-teacher model often outperforms or matches multiple-teacher models across most tasks. Additionally, incorporating advanced optimization techniques further enhances the model's performance. These findings support our idea that "choosy babies need one coach".

1 Introduction

One important feature of child language learning is its incrementality, gradually moving from simple to more complex language. When talking to a child, adults often choose to use simple words and expressions, effectively allowing the child to first focus on what's easy to learn (e.g., Cameron-Faulkner et al., 2003). In machine learning, this 'starting small' approach (Elman, 1991) has informed the paradigm of curriculum learning (Bengio et al., 2009), where models are trained using examples of increasing difficulty.

In the 1st BabyLM Challenge, organized in 2023 to stimulate training of language models on smaller-sized and child-appropriate data sets (Warstadt et al., 2023), curriculum learning was the most commonly used method among all submissions (e.g., Chobey et al., 2023; Martinez et al., 2023; DeBenedetto, 2023). Interestingly, despite its popularity, curriculum learning did not yield consistent improvements over baselines (Warstadt et al., 2023). This suggests that while curriculum learning remains a valuable approach, other methods such as knowledge distillation and architectural modifications may offer additional advantages in certain contexts (Samuel, 2023; Timiryasov and Tastet, 2023, etc.). In our submission to the 2nd BabyLM Challenge, we leverage and combine some of the last year's successful approaches, while also enabling the learner to use a more selective learning strategy.

More specifically, we take as starting point the Baby Llama model and its teacher–student knowledge distillation framework (Timiryasov and Tastet, 2023). We then experiment with changing its objective function from forward KL divergence to reverse KL divergence, inspired by Gu et al. (2024); Agarwal et al. (2024), and implement several strategies to optimize the distillation process. Unlike forward KL, which encourages the student model to approximate the full output distribution of the teacher and often leads to 'mass-covering' behavior, reverse KL focuses on high-probability outputs, helping the student to capture the teacher's main modes. This effectively results in a more selective, or 'choosy' learner. Furthermore, while the original Baby Llama model was trained using an ensemble of two different teacher models, we demonstrate that having a single teacher is sufficient in our setup, which further speeds up the training process and leads us to observe that **choo**sy **ba**bies only need to be trained by one **ca**ch (ChooBaCa).

2 Methodology

The general approach that leads to the development of ChooBaCa is knowledge distillation. We start from Baby Llama as a backbone model (Timiryasov and Tastet, 2023) and implement three important modifications. First, we change the stu-

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 95–105
November 15-16, 2024 ©2024 Association for Computational Linguistics

dent's objective function to reverse KL divergence, following Gu et al. (2024); Agarwal et al. (2024). Second, we replace Baby Llama's ensemble of two teachers with a single teacher, the original LLaMA model (Touvron et al., 2023). Third, we implement several techniques to stabilize the distillation process, inspired by the MiniLLM model of Gu et al. (2024). In the remainder of this section, we unpack the general framework and each of our implemented modifications.

2.1 Distillation framework

We employ a student–teacher distillation setup largely inspired by the Baby Llama model (Timiryasov and Tastet, 2023). Our framework consists of a larger teacher model and a smaller student model, both based on the LLaMA architecture (Touvron et al., 2023). The student model aims to learn the distribution of the teacher model by minimizing the reverse KL divergence between them.

In our setup, the distillation loss, $\mathcal{L}_{\text{distillation}}$, is computed using the reverse KL divergence between the student distribution q_θ and the mixed distribution p_{mixed} (see next section for more details):

$$\mathcal{L}_{\text{distillation}} = T^2 \sum_{i=1}^{N} \sum_{t=1}^{L} q_\theta^{(i,t)} \log \left(\frac{q_\theta^{(i,t)}}{p_{\text{mixed}}^{(i,t)}} \right) \quad (1)$$

where T is the temperature parameter, N represents the batch size, L is the sequence length, $i = 1, 2, \ldots, N$ is the sample batch index, and $t = 1, 2, \ldots, L$ is the time step index within each sequence. The scaling by T^2 compensates for the effect of temperature scaling on the gradients, allowing for more stable optimization.

2.2 Reverse KL divergence

As an alternative to the forward KL divergence objective used for distilling the teachers' knowledge to the Baby Llama model, we use the reverse KL divergence.

Forward KL divergence, KL$[p \parallel q]$, encourages the student model to fit the entire teacher distribution, including low-probability regions. This can lead to mode-averaging (or mass-covering) behavior, where the student assigns unnecessary probability mass to less important areas of the distribution, often resulting in poorer text generation quality.

The reverse divergence, KL$[q \parallel p]$, is commonly used in imitation learning (e.g., Uchibe and Doya, 2021; Ke et al., 2021) and Bayesian methods such as variational inference (see, e.g., Barber, 2012). In the context of knowledge distillation, this objective has been proposed as an alternative to the forward KL divergence (Agarwal et al., 2024; Gu et al., 2024) thanks to its ability to induce mode-seeking behavior, where the student model focuses on the high-probability modes of the teacher model's distribution. This allows the student to capture the key patterns offered by the teacher while ignoring low-probability regions, often less critical for task performance. While this strategy can negatively impact the *diversity* of texts generated by the learner, it is sometimes associated with higher text *quality* (Wiher et al., 2022), which makes it particularly useful for small models, such as Baby Llama, where resource efficiency and accurate learning from limited data are crucial.

Previously, Gu et al. (2024) demonstrated the success of this strategy in instruction-following and long-text generation tasks. Similarly, Agarwal et al. (2024) proposed an on-policy knowledge distillation framework that treats distillation as an imitation learning process, ensuring that the student learns from sequences it is likely to produce during inference. Building upon these insights, we adopt reverse KL divergence in our distillation framework.

2.3 Using a single teacher

While reverse KL divergence effectively concentrates on the teacher's primary modes in single-teacher distillation, challenges arise when this approach needs to be extended to multi-teacher scenarios. Specifically, in such scenarios, the outputs from different teachers can superimpose in potentially conflicting ways. When the student model minimizes the reverse KL divergence across multiple teacher distributions, it may struggle to align with the primary modes due such conflicting signals. As a result, the student model's performance may degrade because it cannot effectively capture the essential modes of individual teachers. Therefore, in our model distillation setup we use a single teacher. Choosing between the two original Baby Llama's teachers, LLaMA and GPT-2, we decided to use LLaMA, as it has the same architecture as the student model.

2.4 Additional optimization techniques

As mentioned above, our use of KL divergence is inspired by Gu et al. (2024), who additionally present several strategies to improve the distillation process in their MiniLLM model. We build up on these strategies and implement the following techniques in our ChooBaCa model, see Appendix A for more details.

Mixing teacher and student outputs. To stabilize training and enhance performance, we mix the logits of the teacher and student models with a mixing coefficient β:

$$z_{\text{mixed}} = \beta z_{\text{teacher}} + (1-\beta) z_{\text{student}} \tag{2}$$

Using this mixture allows the student to benefit from the teacher's knowledge while also incorporating its own learning. This results in a smoother optimization and prevents overfitting to the teacher's distribution.

Single-step decomposition. This is the strategy proposed by Gu et al. (2024), and we adopt it in some of our models. The technique rewrites the gradient calculation to focus on the generation quality of each individual token, rather than accumulating error across the entire sequence. By directly computing the gradient for each token step, it reduces training variance and accelerates convergence, making the optimization process more stable.

Step-wise loss computation. Inspired by the single-step decomposition strategy, we implement a step-wise loss computation technique. Instead of computing the distillation loss over the entire sequence at once, we partition the sequence into smaller segments of length k and calculate the loss for each segment independently. This reduces memory consumption and accelerates training without affecting model performance (Devlin et al., 2019). While single-step decomposition focuses on minimizing variance and improving gradient precision, our step-wise method is primarily designed to prioritize computational efficiency. Additionally, it may help balance gradient flow and adjust errors at finer granularity, making it effective for handling sequences under constrained resources.

Progressive training strategy. The mixing coefficient β descried above can be made dynamic – i.e., it progressively adjusts during training. Initially, the student model heavily relies on the teacher's guidance, but as training progresses, β decreases, allowing the student to become more independent. Specifically, β is updated at each epoch e as follows:

$$\beta_e = \max\left(0.1,\ \beta_{\text{start}} \times \left(1 - \frac{e}{|E|}\right)\right) \tag{3}$$

where β_{start} is the initial value of the mixing coefficient, which is set to 0.7 in our experiments, e is the current epoch number during training, and $|E|$ is the total number of training epochs. Additionally, β is bounded below by 0.1 to prevent it from becoming too small. This progressive strategy helps the student model transition from imitation to autonomous learning, improving generalization (Gou et al., 2021; Mobahi et al., 2020).

The four described strategies enhance the distillation process by stabilizing training, improving efficiency, and enabling the student to effectively learn from the teacher model. By progressively reducing reliance on the teacher, the student model can better generalize from limited data, which is crucial in settings like the BabyLM Challenge.

2.5 Simulation setup

As a backbone architecture, we adopt the 58M parameter version of Baby Llama, optimized for the BabyLM Challenge tasks (Timiryasov and Tastet, 2023). Unless specified otherwise, all the experimental settings, including hyperparameters, dataset splits, and evaluation procedures, strictly follow those outlined in the original study (Timiryasov and Tastet, 2023).

We train and test 12 model variants, summarized in Table 1. The models differ on several dimensions as specified below.

Objective function: reverse KL divergence (as proposed in our study) vs. forward KL divergence (as in the original Baby Llama model).

Number of teachers: one (i.e., LLaMA model, which we expect to be a better fit to our setup) vs. two (i.e., LLaMA and GPT-2, as in the original Baby Llama study).

Data set: the 2nd BabyLM Challenge data set (2024, which is somewhat different from the last year's data set, see Choshen et al., 2024) vs. the 1st BabyLM Challenge data set (2023, as in the original Baby Llama model).

Additional optimization techniques: these are described in Section 2.4, and our model variants differ in terms of the exact subset of techniques they use. Table 1 provides the exact specification for each model variant.

We use the following notation to specify each model variant: [MODEL]-[OBJECTIVE]-[NUMBER OF TEACHERS]-[DATA SET]. For example, CHOOBACA-RV-1-24 is our proposed model with reverse KL divergence and one teacher trained on

No.	Model	Obj.	Tchrs	Data	Additional techniques: Mixing outputs	Single step	Stepwise loss	Progr. training
1	ChooBaCa-fw-2-23	forward	2	2023	–	–	–	–
2	ChooBaCa-fw-1-23	forward	1	2023	–	–	–	–
3	ChooBaCa-fw-2-24	forward	2	2024	–	–	–	–
4	ChooBaCa-fw-1-24	forward	1	2024	–	–	–	–
5	ChooBaCa-rv-2-23	reverse	2	2023	+	+	–	–
6	ChooBaCa-rv-1-23	reverse	1	2023	+	+	–	–
7	ChooBaCa-rv-2-24	reverse	2	2024	+	+	–	–
8	ChooBaCa-rv-1-24	reverse	1	2024	+	+	–	–
9	ChooBaCa-rv-2-23+	reverse	2	2023	+	–	+	+
10	ChooBaCa-rv-1-23+	reverse	1	2023	+	–	+	+
11	ChooBaCa-rv-2-24+	reverse	2	2024	+	–	+	+
12	ChooBaCa-rv-1-24+	reverse	1	2024	+	–	+	+

Table 1: Models used in the experiments. Row 1 corresponds to the original Baby Llama architecture, row 8 is our submission for the 2nd BabyLM Challenge, and rows 9–12 introduce additional optimization techniques that further improve our submission.

the 2024 data set, while ChooBaCa-fw-2-23 is a replication of the original Baby Llama model presented by Timiryasov and Tastet (2023). The model variants whose names end with a '+' suffix (e.g., ChooBaCa-rv-1-24+) introduce additional optimization techniques as specified in Table 1. Detailed experimental settings and configurations can be found in Appendix A.

2.6 Evaluation benchmarks

The 2nd BabyLM Challenge adopts three benchmarks commonly used for evaluating language models.

BLiMP (Benchmark of Linguistic Minimal Pairs, Warstadt et al., 2020) is designed to test models on a variety of syntactic phenomena through pairs of sentences that differ in their grammatical acceptability, providing insight into a model's linguistic capabilities.

GLUE (General Language Understanding Evaluation, Wang et al., 2018) is a suite of tasks for evaluating language understanding, covering areas like sentiment analysis, natural language inference, and semantic similarity. SuperGLUE (Wang et al., 2019) extends GLUE with a more challenging set of tasks, such as causal reasoning, coreference resolution, and question answering, to better benchmark models' advanced comprehension and robustness across diverse linguistic skills.

EWoK (Elements of World Knowledge, Ivanova et al., 2024) is a recently developed benchmark that tests models' factual world knowledge, assessing how well models can apply general knowledge beyond syntactic or linguistic patterns to answer questions about real-world situations.

3 Results and Discussion

Tables 2 and 3 present the evaluation results for all our model variants, as well as the two baselines adopted in the 2nd BabyLM Challenge (the original Baby Llama model and LTG-BERT), on the three benchmarks used in BabyLM (see previous section).

Whereas it is clear from the tables that there is no single best model, we can still observe several important patterns. Our primary finding demonstrates that under reverse KL divergence (see rv models), knowledge distillation with a single-teacher model generally outperforms or matches the performance of models with two teachers. Specifically, within the (Super)GLUE benchmark (11 tasks), rv models with a single teacher outperform two-teacher rv models in 3 tasks (27%) and match their performance in 8 tasks (73%). Within the other two benchmarks – EWoK (11 tasks) and BLiMP (17 tasks), single-teacher rv models match the performance of two-teacher rv models across all tasks. These results support our hypothesis that a choosy, mode-seeking learning strategy enhances the ChooBaCa model's ability to generalize effec-

	Baseline		1	2	3	4	5	6	7	8	9	10	11	12
Task	Baby Llama	LTG-BERT	ChooBaCa-fw-2-23	ChooBaCa-fw-1-23	ChooBaCa-fw-2-24	ChooBaCa-fw-1-24	ChooBaCa-rv-2-23	ChooBaCa-rv-1-23	ChooBaCa-rv-2-24	ChooBaCa-rv-1-24	ChooBaCa-rv-2-23+	ChooBaCa-rv-1-23+	ChooBaCa-rv-2-24+	ChooBaCa-rv-1-24+
(Super)GLUE														
CoLA (MCC)	2.2	0.0	−0.3	4.1	6.3	−5.5	3.0	**22.8**	2.2	6.3	5.0	7.8	14.3	18.2
SST-2	86.2	85.1	86.3	**86.9**	75.5	73.7	84.6	86.0	75.4	75.7	86.3	84.5	77.2	77.8
MRPC (F_1)	82.0	82.2	80.9	80.9	80.1	79.9	80.7	80.9	81.5	81.8	81.2	**82.4**	80.6	79.2
QQP (F_1)	**83.6**	34.2	83.4	82.8	76.8	75.7	82.3	83.4	76.2	75.9	82.0	81.9	79.3	82.9
MNLI	72.4	68.9	**72.7**	71.2	67.3	66.9	71.8	71.4	66.6	67.4	70.6	70.1	68.9	71.0
MNLI-mm	**74.2**	68.9	72.3	72.5	69.0	72.0	72.3	72.4	66.3	67.3	71.7	71.5	71.3	70.9
QNLI	**82.8**	76.5	80.3	80.8	79.0	78.3	79.9	80.7	76.9	76.0	78.4	76.0	77.3	80.6
RTE	49.6	**58.3**	46.0	54.7	55.4	53.7	52.5	46.8	52.5	52.5	51.8	55.4	54.6	56.8
BoolQ	65.0	**68.8**	65.7	66.3	64.0	62.9	66.9	63.4	63.5	62.0	64.2	67.2	62.4	65.5
MultiRC	60.1	58.5	60.1	61.8	65.1	61.2	62.2	61.1	65.2	**65.4**	60.9	63.0	62.1	60.0
WSC	38.5	61.5	48.7	**67.3**	59.6	59.2	57.7	38.5	61.5	61.4	55.7	38.4	62.0	63.4
EWoK														
Social interactions	50.7	51.7	50.3	50.3	51.7	50.0	51.7	51.7	52.4	50.0	51.3	**52.7**	51.0	51.0
Physical relations	50.6	51.0	50.4	50.4	48.9	49.8	51.1	50.4	50.9	50.0	51.0	**51.4**	47.6	51.1
Spatial relations	46.7	45.3	49.4	50.0	47.1	48.7	49.8	50.0	47.2	49.6	49.6	49.8	**50.2**	48.9
Material properties	49.4	45.3	47.7	49.4	49.4	49.4	48.2	47.7	47.1	49.4	48.8	**50.6**	48.8	47.6
Agent properties	50.5	50.2	50.4	50.2	50.0	49.9	50.4	50.6	50.7	49.8	49.6	50.3	50.0	**51.2**
Material dynamics	51.7	51.0	50.8	50.4	51.2	50.9	50.9	51.3	49.2	50.9	49.1	49.6	53.0	**55.5**
Physical dynamics	54.2	42.5	50.8	50.8	50.8	49.2	50.8	50.8	52.5	50.8	49.2	**54.2**	49.1	52.5
Physical interaction	50.4	49.1	**51.4**	50.7	50.5	50.5	**51.4**	50.9	49.6	50.5	48.9	**51.4**	49.4	51.0
Social properties	50.3	53.4	49.1	49.1	49.1	48.8	50.6	50.3	47.9	49.7	**53.0**	49.4	50.3	50.3
Quantitative properties	53.5	48.1	52.2	**55.7**	53.5	51.2	53.1	51.6	52.9	53.2	50.7	53.5	52.8	51.9
Social relations	49.8	50.6	50.1	**50.9**	50.5	50.1	50.5	50.1	49.7	50.1	50.2	50.4	49.5	49.8
BLiMP														
Anaphor Agr.	**92.1**	81.3	84.4	86.0	87.5	80.7	88.9	82.8	89.8	91.2	84.7	86.4	89.4	91.9
Arg. Structure	73.7	56.8	68.4	71.1	68.6	65.6	69.7	71.2	75.1	**75.2**	69.6	68.9	73.9	72.5
Binding	71.1	68.2	71.7	**75.1**	71.4	69.3	70.5	68.9	69.1	60.4	71.8	72.4	69.7	71.5
Control/Raising	**67.2**	48.5	67.4	65.3	65.5	60.9	67.0	67.1	60.3	57.2	65.1	62.5	58.6	56.6
Det.-Noun Agr.	87.0	77.6	88.6	91.7	87.8	83.8	91.5	**91.6**	89.8	88.9	89.4	88.8	87.7	87.4
Ellipsis	69.7	43.8	67.8	67.9	70.8	58.4	69.1	68.3	68.1	**72.4**	65.4	64.2	66.9	69.9
Filler-Gap	70.1	66.8	59.4	58.8	**70.9**	54.9	56.6	65.9	70.1	65.3	60.2	61.3	52.3	62.2
Irregular Forms	85.3	59.8	**92.3**	83.4	74.1	84.3	90.1	82.5	86.0	81.5	85.7	83.3	82.9	84.4
Island Effects	50.5	45.8	48.2	50.5	54.1	48.8	46.7	49.4	54.0	**59.2**	47.1	44.6	44.9	50.4
NPI Licensing	50.8	**68.2**	48.5	52.2	52.6	42.5	51.1	51.3	37.5	43.2	45.2	53.2	61.9	37.3
Quantifiers	76.4	44.2	64.9	58.5	81.1	60.1	75.3	71.7	65.0	71.6	61.6	76.7	**77.0**	73.6
Subj.-Verb Agr.	82.3	75.6	82.2	80.4	67.5	62.3	**83.7**	80.7	80.5	78.3	79.8	82.1	79.6	80.5
BLiMP suppl.														
Hypernym	49.6	54.2	46.8	48.9	48.7	**51.1**	48.9	46.8	50.1	48.1	48.4	48.0	48.4	48.9
QA Congruence (easy)	54.7	62.5	46.9	54.7	56.3	45.3	53.1	48.4	54.7	54.7	53.1	51.6	**56.3**	53.1
QA Congruence (tricky)	41.2	**49.1**	35.8	38.8	37.6	36.3	38.8	38.2	43.6	40.0	34.6	37.6	38.8	43.3
Subj.-Aux. Inversion	86.0	79.9	88.0	86.8	87.5	84.4	85.0	82.6	79.1	86.7	**88.2**	81.2	84.5	87.2
Turn Taking	66.1	58.2	59.6	64.6	66.4	54.3	60.0	65.4	66.1	66.7	63.2	63.6	65.0	**67.1**

Table 2: SuperGLUE, EWoK, and BLiMP evaluation results (zero-shot accuracy, unless specified otherwise) for various variants of ChooBaCa and the baselines.

Model	BLiMP	BLiMP-suppl.	EWoK	GLUE	Macroaverage
CHOOBACA-FW-2-23	68.1	55.4	50.2	66.5	60.1
CHOOBACA-FW-1-23	68.7	58.7	50.7	**70.2**	**62.8**
CHOOBACA-FW-2-24	**70.2**	59.3	50.2	66.9	61.7
CHOOBACA-FW-1-24	63.6	53.2	49.9	66.0	58.2
CHOOBACA-RV-2-23	69.4	57.0	50.8	68.5	61.4
CHOOBACA-RV-1-23	68.3	56.3	50.5	65.0	60.0
CHOOBACA-RV-2-24	69.3	59.5	50.0	66.0	61.2
CHOOBACA-RV-1-24	69.0	58.7	50.4	66.0	61.0
CHOOBACA-RV-2-23+	68.0	57.5	50.2	67.5	60.8
CHOOBACA-RV-1-23+	68.4	56.4	**51.2**	65.8	60.5
CHOOBACA-RV-2-24+	69.5	58.6	50.2	67.0	61.3
CHOOBACA-RV-1-24+	68.0	**59.9**	51.0	68.3	61.8
BABY LLAMA	69.8	59.5	50.7	63.3	60.8
LTG-BERT	60.6	60.8	48.9	60.3	57.7

Table 3: Aggregated evaluation results of ChooBaCa model variants and baseline models across all benchmarks.

tively across diverse language understanding tasks under the reverse KL divergence framework.

When comparing reverse KL divergence models (RV) to forward KL divergence models (FW), our results indicate that RV models achieve better or comparable performance in the majority of tasks. Specifically, when trained on the 2024 data set, across the SuperGLUE and EWoK benchmarks (22 tasks), RV models with a single teacher outperform FW models with two teachers in 3 tasks (14%) and match their performance in 18 tasks (81%), with FW models slightly outperforming RV models in 1 task (5%). This result highlights the effectiveness of inducing mode-seeking behavior through reverse KL divergence, as RV models focus on high-probability linguistic patterns, leading to improved generalization and performance across various benchmarks compared to the traditional FW approach with two teachers.

Models that incorporate additional optimization techniques (marked with a '+' character at the end) show better performance under the reverse KL divergence setting. These models outperform their non-optimized counterparts in 8 out of 22 (Super)GLUE and EWoK tasks (36%) and 7 out of 17 BLiMP tasks (41%). In the remaining tasks, their performance is comparable. Similarly, when trained on the 2023 dataset, optimized RV models show improvements in 5 out of 22 (Super)GLUE and EWoK tasks (23%) and 4 out of 17 BLiMP tasks (24%), with performance remaining comparable in most other tasks. These findings suggest that the additional optimization techniques, namely step-wise loss computation and progressive training strategy, contribute to a more stable and efficient training processes, enabling the models to better capture complex linguistic structures.

Comparing the results for models trained on the 2023 vs. the 2024 dataset, we observe consistent patterns across the two. When trained on the 2023 dataset, RV models with a single teacher outperform two-teacher RV models in 12 out of 22 (Super)GLUE and EWoK tasks (55%) and 8 out of 17 BLiMP tasks (47%), with performance being comparable in 7 (Super)GLUE and EWoK tasks (32%) and 2 BLiMP tasks (12%). These patterns across both datasets once again support our main argument: choosy babies need one coach.

Notably, the performance of all models on the EWoK benchmark is close to chance level, 50%. This suggests that our relatively small models might lack the capacity to effectively handle the complex EWoK tasks, which are likely more demanding compared to the other benchmarks included in this study.

Overall, these results suggest that a selective, mode-seeking learning strategy, based on the use of reverse KL divergence with a single teacher model, enhances the ChooBaCa model's ability to generalize effectively across diverse language understanding tasks. At the same time, all ChooBaCa variants struggle with more complex tasks grounded in real-world knowledge.

4 Conclusion

Our findings support the use of reverse Kullback–Leibler divergence in knowledge distillation, particularly in a single-teacher setup. While it has been shown that multiple instructional sources can be advantageous (Timiryasov and Tastet, 2023; Odumakinde et al., 2024), our results suggest that in a constrained setup with one small model trained on limited amounts of data, in combination with using reverse KL divergence, an ensemble of teachers may not be necessary. Our single-teacher setup promotes mode-seeking behavior, resulting in a more focused and efficient learning process. It also simplifies the learning process and eliminates the need to train more than one teacher model. Our ChooBaCa model is able to efficiently generalize across diverse language understanding tasks.

In future work, we plan to explore hybrid KL divergence methods, such as alternating between forward and reverse KL divergence or employing a weighted combination during training, to balance the learner's focus between dominant and minor modes. Additionally, investigating layer-wise distillation – where different layers of the student model learn from different teachers – could more effectively accommodate varied distribution peaks. Finally, we aim to examine dynamically averaging the outputs of multiple teachers before applying reverse KL divergence, which might smooth out the distribution and help the student model identify and prioritize significant modes without bias.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*.

David Barber. 2012. *Bayesian reasoning and machine learning*. Cambridge University Press.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48.

Thea Cameron-Faulkner, Elena Lieven, and Michael Tomasello. 2003. A construction based analysis of child directed speech. *Cognitive Science*, 27:843–873.

Aryaman Chobey, Oliver Smith, Anzi Wang, and Grusha Prasad. 2023. Can training neural language models on a curriculum with developmentally plausible data improve alignment with human reading behavior? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 98–111. Association for Computational Linguistics.

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*.

Justin DeBenedetto. 2023. Byte-ranked curriculum learning for BabyLM strict-small shared task 2023. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 198–206. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Jeffrey L Elman. 1991. Incremental learning, or the importance of starting small. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 13.

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

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

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 Akyürek, 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. *Preprint*, arXiv:2405.09605.

Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. 2021. Imitation learning as f-divergence minimization. In

Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14, pages 313–329. Springer.

Richard Diehl Martinez, Hope McGovern, Zebulon Goriely, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – curriculum learning for infant-inspired model building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 112–127. Association for Computational Linguistics.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. 2020. Self-distillation amplifies regularization in hilbert space. *Advances in Neural Information Processing Systems*, 33:3351–3361.

Ayomide Odumakinde, Daniel D'souza, Pat Verga, Beyza Ermis, and Sara Hooker. 2024. Multilingual arbitrage: Optimizing data pools to accelerate multilingual progress. *arXiv preprint arXiv:2408.14960*.

David Samuel. 2023. Mean BERTs make erratic language teachers: the effectiveness of latent bootstrapping in low-resource settings. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 221–237. Association for Computational Linguistics.

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. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Eiji Uchibe and Kenji Doya. 2021. Forward and inverse reinforcement learning sharing network weights and hyperparameters. *Neural Networks*, 144:138–153.

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 Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, et al. 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*.

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.

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012.

A Appendix

A.1 Optimization methods

In this appendix, we provide detailed explanations of the optimization methods and formulas used in our approach, including definitions of all symbols.

A.1.1 Progressive distillation strategy

To enhance the distillation process and allow the student model to gradually become more independent from the teacher, we introduce a dynamic mixing coefficient β that progressively reduces the teacher's influence during training. β starts with a higher value and decreases as training progresses, ensuring that the student model relies more on the teacher's guidance at the beginning of training and gradually becomes more autonomous.

A.2 Reverse KL divergence

A.2.1 Loss function modification

In standard knowledge distillation, the student model learns by minimizing the forward Kullback–Leibler (KL) divergence between the teacher's output distribution and the student's output distribution:

$$\mathcal{L}_{\text{F-KL}} = \mathrm{KL}\left(P_{\text{teacher}} \parallel P_{\text{student}}\right) \tag{4}$$

However, to induce mode-seeking behavior in the student model, we instead minimize the reverse KL divergence:

$$\mathcal{L}_{\text{R-KL}} = \mathrm{KL}\left(P_{\text{student}} \parallel P_{\text{teacher}}\right) \tag{5}$$

where P_{teacher} is the probability distribution over the output tokens from the teacher model, P_{student} is the same distribution from the student model, and $\mathrm{KL}(P \parallel Q)$ is the Kullback–Leibler divergence from distribution P to distribution Q.

Minimizing the reverse KL divergence encourages the student model to focus on the high-probability regions (modes) of the teacher's distribution.

A.3 Implementation details

Mixing teacher and student logits. To stabilize training and facilitate the progressive distillation strategy, we mix the logits (pre-softmax outputs) from the teacher and student models. The mixed logits z_{mixed} are computed using the dynamic mixing coefficient β. This approach ensures a smooth transition for the student model from relying on the teacher to developing its own understanding.

Temperature Scaling. We apply temperature scaling to the logits to soften the probability distributions and make them more suitable for distillation. The scaled logits are:

$$\tilde{z}_{\text{student}} = \frac{z_{\text{student}}}{T} \tag{6}$$

$$\tilde{z}_{\text{mixed}} = \frac{z_{\text{mixed}}}{T} \tag{7}$$

where T is the temperature parameter (we set $T = 2.0$ in our experiments). Higher temperatures produce softer probability distributions.

Computing probability distributions. We compute the probability distributions using the softmax function:

$$\begin{aligned} q_\theta &= \text{softmax}\left(\tilde{z}_{\text{student}}\right) \\ p_{\text{mixed}} &= \text{softmax}\left(\tilde{z}_{\text{mixed}}\right) \end{aligned} \tag{8}$$

where q_θ and p_{mixed} are, respectively, the student's and the mixed teacher–student probability distributions after temperature scaling.

Loss computation. The distillation loss $\mathcal{L}_{\text{distillation}}$ is computed using the reverse KL divergence between the student distribution and the mixed teacher-student distribution. The scaling by T^2 compensates for the effect of temperature scaling on the gradients, allowing for more stable optimization.

Total loss. The total loss $\mathcal{L}_{\text{total}}$ combines the standard cross-entropy loss on the student model's outputs and the distillation loss:

$$\mathcal{L}_{\text{total}} = \alpha \cdot \mathcal{L}_{\text{student}} + (1 - \alpha) \cdot \mathcal{L}_{\text{distillation}} \tag{9}$$

where $\mathcal{L}_{\text{student}}$ is the cross-entropy loss between the student model's predictions and the ground truth tokens, and α is a weighting factor (we set $\alpha = 0.5$ in our experiments).

Cross-entropy loss. The student loss $\mathcal{L}_{\text{student}}$ is computed as:

$$\mathcal{L}_{\text{student}} = \frac{1}{N} \sum_{i=1}^{N} \ell_{\text{CE}}(q_\theta(y_i|x_i), y_i) \tag{10}$$

where:

- ℓ_{CE} is the cross-entropy loss function.
- $q_\theta(y_i|x_i)$ is the student model's predicted probability distribution for the target token y_i given input x_i.
- y_i is the ground truth token.

Step-wise loss computation. To improve computational efficiency and reduce memory usage, we compute the distillation loss over smaller chunks of the sequence. Specifically, we divide the sequence into segments of length k (we use $k = 5$ in our experiments) and compute the loss for each segment separately. This step-wise computation allows us to handle longer sequences without exceeding memory limitations.

A.4 Optimization and training setup

A.4.1 Optimizer and learning rate scheduler

We use the AdamW optimizer with the following hyperparameters:

- Learning rate: $\eta = 2.5 \times 10^{-4}$
- Betas: $\beta_1 = 0.9$, $\beta_2 = 0.999$
- Epsilon: $\epsilon = 1 \times 10^{-8}$
- Weight decay: $\lambda = 0.01$

We employ a cosine annealing learning rate scheduler with a maximum number of iterations $T_{\text{max}} = 500$.

A.4.2 Training hyperparameters

The training hyperparameters are set as follows:

- Batch size: $N = 32$
- Sequence length: $L = 128$
- Number of epochs: $E = 6$
- Gradient accumulation steps: $G = 1$
- Mixed-precision training: FP16

A.5 Summary of notations

For clarity, we summarize the notations used in our formulas:

- z_{teacher}: logits from the teacher model.
- z_{student}: logits from the student model.
- z_{mixed}: mixed logits from teacher and student.
- β: dynamic mixing coefficient.
- T: temperature parameter for scaling logits.
- $\tilde{z}_{\text{student}}$: temperature-scaled student logits.
- $\tilde{z}_{\text{mixed}}$: temperature-scaled mixed logits.
- q_θ: student model's probability distribution after temperature scaling.
- p_{mixed}: mixed probability distribution after temperature scaling.
- $\mathcal{L}_{\text{student}}$: cross-entropy loss between student predictions and ground truth.
- $\mathcal{L}_{\text{distillation}}$: distillation loss computed using reverse KL divergence.
- $\mathcal{L}_{\text{total}}$: total loss combining student loss and distillation loss.
- α: weighting factor between student loss and distillation loss.
- N: batch size.
- L: sequence length.
- k: chunk size for step-wise loss computation.
- E: number of training epochs.
- e: current epoch number during training
- η: learning rate.
- β_1, β_2: beta parameters for AdamW optimizer.
- ϵ: epsilon parameter for AdamW optimizer.
- λ: weight decay parameter.
- T_{max}: maximum number of iterations for cosine annealing scheduler.
- G: gradient accumulation steps.

A.6 Code implementation

The methods described above are implemented in our code, which we make publicly available[1]. The code includes the implementation of the progressive distillation strategy, reverse KL divergence loss computation, mixing of teacher and student logits, and the optimization setup with the AdamW optimizer and cosine annealing scheduler.

A.7 Efficiency enhancements

To improve computational efficiency, we compute the distillation loss over chunks of $k = 5$ tokens. This step-wise loss computation reduces memory consumption and accelerates training without compromising performance.

A.8 Algorithm summary

Combining all the components, the training algorithm operates as follows:

1. **Initialize** the student model parameters θ, mixing coefficient β_{start}, temperature T, and weighting factor α.
2. **For** each epoch $e = 1$ to E:
   (a) Update β.
   (b) **For** each mini-batch:
      i. Compute student logits z_{student}.
      ii. Compute teacher logits z_{teacher} (with no gradient computation).
      iii. Compute mixed logits z_{mixed}.
      iv. Scale logits with temperature T.
      v. Compute probability distributions q_θ and p_{mixed}.
      vi. Compute $\mathcal{L}_{\text{student}}$ using cross-entropy loss.
      vii. Compute $\mathcal{L}_{\text{distillation}}$ using reverse KL divergence.
      viii. Compute total loss $\mathcal{L}_{\text{total}}$.

[1] https://github.com/todamoonnback/ChooBaCa

ix. Backpropagate gradients and update model parameters using AdamW optimizer.

3. **End For**

This algorithm ensures that the student model gradually shifts from relying on the teacher's guidance to developing its own representations, focusing on the high-probability modes of the teacher's distribution.

Different Ways to Forget: Linguistic Gates in Recurrent Neural Networks

Cristiano Chesi[1] **Veronica Bressan**[1] **Matilde Barbini**[1] **Achille Fusco**[1]
Maria Letizia Piccini Bianchessi[1] **Sofia Neri**[1] **Sarah Rossi**[1] **Tommaso Sgrizzi**[1]
[1]NeTS, IUSS Pavia
{cristiano.chesi, veronica.bressan, matilde.barbini, achille.fusco, letizia.piccinibianchessi, sofia.neri, sarah.rossi, tommaso.sgrizzi}
@iusspavia.it

Abstract

This work explores alternative gating systems in simple Recurrent Neural Networks (RNNs) with the intent to induce linguistically motivated biases during training, ultimately affecting models' performance on the BLiMP task. Here we focus on the BabyLM 10M training corpus only (Strict-Small Track). Our experiments reveal that: (i) standard RNN variants—LSTMs and GRUs—are insufficient for properly learning the relevant set of linguistic constraints; (ii) quality and size of the training corpus have little impact on these networks since we observed comparable performance of LSTMs trained exclusively on the child-directed speech portion of the corpus; (iii) increasing the size of the embedding and hidden layers does not significantly improve performance. In contrast, specifically gated RNNs (eMG-RNNs), inspired by certain Minimalist Grammar intuitions, exhibit advantages in both training loss and BLiMP accuracy although their performance is not yet comparable to that of humans.

1 Introduction[1]

Despite their impressive performance, transformers-based architectures (Vaswani et al., 2017) provide limited insight from a theoretical linguistic perspective and tend to perform poorly when trained on small datasets, unless ad-hoc optimizations are applied (Charpentier and Samuel, 2023; Xu et al., 2024). In this paper, we focus on simple recurrent architectures to explore the effect of linguistic biases potentially induced by specific gating systems on both cross-entropy loss and performance in forced-choice tasks such as BLiMP (Warstadt et al., 2020). The goal is to preserve the role of incremental processing, which is obfuscated by the attention mechanism in transformers while retaining the self-supervised (autoregressive) training approach. Such obfuscation arises from the fact that, while human linguistic processing operates in a strictly incremental manner (Bever, 1970), the computation of gradients required to minimize model loss during LLM training must be performed in parallel for computational efficiency. This legitimate pursuit of reducing computational complexity has led, on one hand, to attention-based approaches that operate in parallel on the full input vector, composed of a fixed-length sequence of tokens, and, on the other hand, to simplifications in RNNs—such as removing any time-dependent interaction between the hidden state and the input. This last approach ultimately transfers the computational burden from the inefficient backward propagation through time (BPTT) approach to the need for additional layers (Feng et al., 2024). A relevant challenge to the Poverty of Stimulus hypothesis (Yang et al., 2017) can then be formulated in the following terms: Can a Small Language Model (SLM)—trained with a limited amount of data and under ecological exposure comparable to that of young learners—attain an adult level of linguistic competence (Chomsky, 1965)? From this perspective, linguistic competence is measured simply by the model's performance on each BLiMP subtest: a SLM will be considered *consistent*—i.e., displaying adult-like linguistic competence—if it systematically selects (> 72-80% of the times[2], sentences like "Susan

[1] Preprocessing, tokenization, models' architecture, training procedure and results are available at: https://github.com/cristianochesi/babylm-2024

[2] This is a prudential threshold obtained from the average human performance reported on BLiMP (~88%, Warstadt et al., 2020) minus 1 or 2 standard deviations (~8%).

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 106–117
November 15-16, 2024 ©2024 Association for Computational Linguistics

revealed herself", which satisfy anaphoric binding (Condition A or similar generalization predicting that an anaphor like *herself* must be bound within the relevant domain, Chomsky, 1981) over the minimally different alternative "Susan revealed themselves," despite irrelevant lexical variations. We focus our analysis solely on the BLiMP minimal pair decision task to avoid complex issues related to general acceptability and coherence considerations required to interpret raw probability outputs (Lau et al., 2017).

To preserve the cognitively plausible, albeit computationally inefficient, incremental approach, we adapted Recurrent Neural Network (RNN) models (Elman, 1990) and made minimal modifications to the standard LSTM (Hochreiter and Schmidhuber, 1997) and GRU (Cho et al., 2014) architectures through gating alterations. We recorded the effects of these modifications on training loss and accuracy, and we compared the models' performance on the BLiMP task. For comparison, we also report the performance of well-studied LSTM and GRU architectures (Gulordava et al., 2018; Chowdhury and Zamparelli, 2018) after training on the 10M tokens dataset provided for the BabyLM Challenge 2024 Strict-Small Track. We hypothesized that by modifying the information flow within the network, we could provide architectural scaffolding for C-command constraints, as defined in §3 (Reinhart, 1976). We then considered two distinct computational pathways: one for preserving part of the memory content, whenever an indication that an unsatisfied dependency is present (*Move* gate), and the other for deciding whether to keep expanding the previous constituent—the "sequential phase", Bianchi and Chesi, 2014—or instantiating a nested constituent—embeddings. We then selectively simplify one pathway or the other to measure the impact of these alterations on various structural aspects. The paper is organized as follows: we first present the basic preprocessing steps adopted to clean the 10M-token corpus (§2.1). We then discuss the BLiMP dataset, focusing on the specific grammatical constraints necessary to correctly evaluate the relevant contrasts (§2.2). We conclude the introductory section by discussing the computational graphs representing standard LSTM and GRU architectures, finally speculating on the relevance of certain gating solutions from a linguistic perspective. Section 3 introduces the core linguistic intuitions we aim to model, along with attempts to rephrase these intuitions in simple, albeit potentially simplistic, combinatorial terms. Section 4 describes the basic architecture we tested, dubbed expectation-Based Minimalist Grammar Recurrent Neural Network—eMG-RNN—, loosely inspired by an unorthodox interpretation (Chesi, 2022) of Minimalist Grammars (Stabler, 2013; Chomsky et al., 2023). Section 5 presents the results of our tests, showing that the gating proposals effectively capture certain linguistic constraints but not others. Overall, the performance of eMG-RNNs is higher compared to that of LSTMs and GRUs. More importantly, unlike any LSTM and GRU architecture, eMG-RNNs consistently show biases towards one item of the minimal pairs (73% of the time for the correct item, 27% of the times for the incorrect one) in various phenomena (44% of the BLiMP filtered subtasks). We conclude with a general discussion on how different regimens have impacted these results and outline the next steps toward achieving a more precise implementation of the relevant linguistic biases that remain unresolved in the current experiments.

2 Training data, benchmarks, and RNN architectures

In this section, we present the preprocessing routines we adopted to prepare the training data for our models (§2.1). We then discuss some fundamental linguistic aspects related to the BLiMP task used to assess the linguistic performance of our models (§2.2). Finally, we introduce the standard RNN architectures—LSTM and GRU—used as starting points for our experiments (§2.3).

2.1 Corpus preprocessing

The original corpus provided with the Strict-Small Track of the BabyLM 2024 challenge consists of roughly 10M words. Six different sections are included: child-directed speech from CHILDES (MacWhinney, 2000), movie subtitles from OpenSubtitles (Lison and Tiedemann, 2016), the dialogue portion of the British National Corpus (BNC Consortium, 2007), telephone conversations from the Switchboard Dialog Act Corpus (Godfrey et al., 1992; Stolcke et al., 2000), written English from the Standardized Project Gutenberg Corpus

(Gerlach and Font-Clos, 2020), and from Simple Wikipedia (simplewiki/20221201). Because of similar preprocessing necessities, we grouped together under the label ‘conversations’ the BNC and Switchboard sections. Table 1 reports some details on corpus size and richness (Type-Token Ratio, TTR) before and after preprocessing.

Section	**Before**	**After**
	Tokens (TTR)	
CHILDES	1,920,655 (0.02)	1,913,959 (0.01)
SUBTITLES	2,041,868 (0.06)	2,399,780 (0.02)
CONVERSATIONS	1,079,286 (0.04)	1,211,618 (0.02)
GUTENBERG	2,539,489 (0.05)	2,895,199 (0.02)
WIKIPEDIA	1,453,539 (0.09)	1,546,763 (0.05)
ALL	9,034,837 (0.04)	9,967,319 (0.01)

Table 1: BabyLM 10M Corpus profile.

A uniform preprocessing pipeline is applied across all sections. This step includes converting text to lowercase, normalizing punctuation (e.g., adding spacing around punctuation, splitting lines after strong punctuation), removing extra spaces and line breaks, and preventing the incorrect splitting of abbreviations like *mr.* and *mrs.* by removing the dot after them. We relied solely on punctuation to segment sentences. After processing, the average word per sentence was 9 and 85% of sentences consisted of less than 75 words. Minor adjustments were made to accommodate the specific formatting characteristics of each section. These variations ensured that the preprocessing remained effective and adapted to the unique aspects of the data, while still adhering to a broadly uniform approach. For example, in the CHILDES and Switchboard sections, we removed metalinguistic information (e.g., speaker labels like *A: … B: …* or **CHI:*) and transcription symbols (e.g., *&-*, *&+*). Additionally, we made other minor adjustments specific to the corpus format, such as normalizing quotes, handling acronyms, and removing brackets. The goal of the preprocessing step was to remove any metalinguistic information and retain only the relevant phonological information (essentially pauses and rough intonation as indicated by question and exclamation marks). Obviously, removing speaker labels and converting everything to lowercase significantly undermines the model's performance on the GLUE, BLiMP Supplement, and EWoK tasks. However, as we have stated from the beginning, achieving better performance on these tasks was not our main goal.

2.2 The BLiMP dataset

The Benchmark of Linguistic Minimal Pairs for English (BLiMP, Warstadt et al., 2020) is a test set designed to assess the grammatical knowledge expressed by LLMs in English. It includes 67 groups of phenomena, each consisting of 1,000 minimal pairs of sentences that sharply contrast in grammatical acceptability. The phenomena are categorized into 12 distinct areas, such as anaphor agreement, binding, control/raising, determiner-noun agreement, ellipsis, filler-gap dependencies, irregular forms, and island effects. The pairs are generated using grammatical templates and the estimated individual human agreement with the judgments is 88.6% overall (based on judgments on 100 annotations from each paradigm). N-gram, LSTM, and Transformer language models are evaluated by assessing whether they assign a higher probability to the grammatically correct sentence in each minimal pair. To mitigate issues arising from sentence length differences (as models that sum the log probability of each token may simply penalize longer sentences), the length of the minimal pairs was kept constant. However, this approach limits the minimal contrasts that can be tested. For instance, we cannot infer from the test whether simply filling a proper gap in a wh-question influences the returned probability (e.g., “*who* do you believe X criticized $\_{\text{who}}$?” vs. “*who* do you believe X criticized Y?”). In such cases, the legitimate option adopted in BLiMP is to alternate a *wh-* item like *who* with the complementizer *that,* as in “X figured out *that* Y appreciates Z” vs. "X figured out *who* Y appreciates Z”—see §3.

2.3 Models’ architecture: RNNs strike back

Although RNNs have been largely surpassed by transformers in nearly all NLP tasks, their cognitive transparency remains commendable. A recent resurgence, in the past couple of years, has also shown that both training efficiency and state-of-the-art performance can still be achieved (Feng et al., 2024; Gu and Dao, 2024). The simple idea that learning can be reduced to improving next-token prediction using word-by-word self-supervision is sufficiently ecological in the sense that it fits with the observation that children do not use adults’ supervision in language acquisition (Yang et al., 2017). Importantly, research has demonstrated that pre-trained transformer-based LMs exhibit significant differences from human performance, for instance, in correlation with reading times (Steuer et al., 2023) and in how they handle negation (Ettinger, 2020) and word order (Pham et al., 2021). More generally, the fact that transformers process all words in a sentence simultaneously (using self-attention) does not allow us to capture the incremental processing typical of human language understanding.[3] This process plays in fact a crucial role in the cognitive parsing of syntactic dependences (Frazier, 1987). Linguistic intuition often involves building up meaning incrementally, which RNNs inherently capture through their sequential processing. The LSTM architecture, for instance, centers around a sequence of gates and states that regulate information flow, possibly mirroring some relevant cognitive notion of short-term and long-term memory in language comprehension. This interpretability enables us to gain insights into how linguistic properties are represented and handled. Conversely, the (self-)attention mechanism is more opaque, involving multiple layers of attention heads that can be challenging to interpret from a linguistic perspective. We think it is then important to explore further the gating system at least in LSTM and GRU standard architectures.

2.3.1 LSTM

In all computational graphs that follow, *x* represents the input, *h* the hidden layer—or the main output—, and *c* an additional contextual output—if present; *E* represents the "embedding" consolidation—a simple linear transformation from one-hot encoded input to a lower dimensionality vector. The symbol "⁀" denotes the concatenation operation, while σ and *tanh* refer to the *sigmoid* and *tanh* transformations respectively. ⊙ represents the Hadamard product and + the summation. Adopting these conventions, a standard LSTM network is described in Figure 1. One of the crucial gates in this architecture is the so-called *forget gate*, denoted as *f*. Due to the sigmoid transformation, when the result is multiplied (⊙) by the cell activation c_t, certain components in c_t are deleted, or "forgotten", whenever the *f* activation output values close to 0.

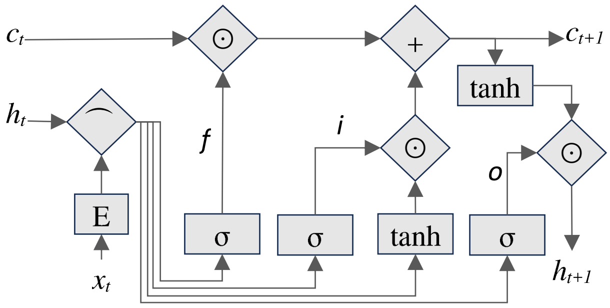


Figure 1: LSTM computational graph

In fact, the intent of the LSTM gating system was exactly to create a double pathway to process, on the one hand, the local contribution of each input component (*i* pathway), on the other, the long-distance contribution of c_t. Linguistically speaking, while the concept of "forgetting" seems transparent and powerful to us, both the formation of the output (*o*) and the contribution of the context to this output—a *tanh* transformation multiplied (⊙) by *o*—seem linguistically too unconstrained: Aside from the sigmoid transformation, a crucial decision must be made based on the simple concatenation of the hidden state and the input in both the *f* and *o* gates.

As far as the BLiMP test is concerned, the performance of the best-performing LSTM architecture—consisting of 650 embedding units (henceforth abbreviated as E650) and 650-units in each of the two hidden layers (henceforth abbreviated as H650) (Gulordava et al., 2018)—trained with 90M tokens from English Wikipedia (Warstadt et al., 2020) is reported in Table 2 below.

	LSTM	***Human***
Overall	68.9	88.6
Ana. agr	91.7	97.5
Arg. str	73.2	90
Binding	73.5	87.3
Ctrl. raising	67	83.9
D-N agr	85.4	92.2
Ellipsis	67.6	85
Filler. gap	72.5	86.9
Irregular	89.1	97
Island	42.9	84.9
Npi	51.7	88.1
Quantifiers	64.5	86.6
S-V agr	80.1	90.9

Table 2. LSTM and Human performance on BLiMP (Warstadt et al., 2020)

[3] On our limited capacity to process tokens "in parallel" one might be interested in the rapid parallel visual presentation (RPVP) task (Snell and Grainger, 2017) and on the relevant restrictions observed during this task (Fallon and Pylkkänen, 2024).

Kuncoro et al. (2018), among others, examined the impact of incorporating syntactic information into LSTM models, using syntax-sensitive dependencies like subject-verb agreement. They adapted Recurrent Neural Network Grammars (RNNGs), which utilize hierarchical phrase-structure trees, and found that while LSTMs can learn syntax-sensitive dependencies when given sufficient capacity, their accuracy declines as the number of attractors increases due to a bias toward more recent sequential information. RNNGs, which explicitly model syntactic structures through hierarchical representations, performed better than LSTMs, highlighting the importance of how syntactic structures are integrated into a model.

2.3.2 GRU

Gated recurrent units (Cho et al., 2014) can be interpreted as simplified LSTM networks that avoid storing information on an independent context output and attempt to control non-local information by means of a clever Update gate (*u*), as illustrated in Figure 2.

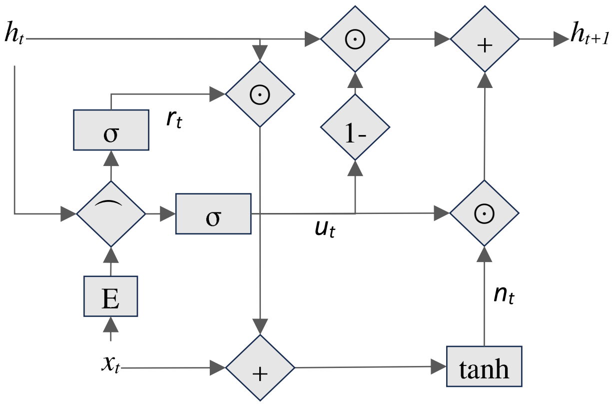


Figure 2: GRU computational graph

The output of each cell (*h*) is conditioned by the contribution of the update gate (*u*): the higher *u*, the greater the contribution of the previous hidden activation passed through the new gate information (*n*)—this is because *h* is multiplied (⨀) by 1-*u*; the lower *u*, the greater the contribution of the—modified—previous activation *h*—the input elaboration is simply multiplied (⨀) by *u*. The linguistic interest for this specific mechanism will be explained in the next section.

3 Core linguistic constraints and gates

According to Minimalism (Chomsky et al., 2023), *Merge* (*M*) is the fundamental structure-building operation. It is recursive, in the sense it applies to the result of other Merge operations, it is binary since it always takes two arguments, and it is local since the elements that Merge must be adjacent. Suppose *a*, *b*, *c*, *d*, and *e* are lexical items, then:

(1) *M*(*M*(*e*, *M*(*c*, *d*)), *M*(*a*, *b*))) =
{{*e* {*c d*}} {*a b*}}

That is, the structure obtained after the application of four *M* operations in this exact order—i. *M*(*a*, *b*), ii. *M*(*c*, *d*), iii. *M*(*e*, {*c d*}), iv. *M*({*e* {*c d*}}, {*a b*})—yields a nested constituent {*c d*} that cannot enter into any relevant structural relation with the constituent {*a b*} (e.g., anaphor binding, as in "[$_e$ the patient$_j$ [$_c$ of [$_d$ the doctor$_i$]]] [$_a$ blames [$_b$ himself$_{*i/j}$]]"). Although this approach offers significant descriptive advantages, little attention has been given to how structure-building operations can be executed in real-time. A relatively lively debate suggests that real-time considerations may be important, both supporting behavioral evidence (Zaccarella and Friederici, 2015; Chesi and Canal, 2019) and computational predictions (Kobele et al., 2013). Especially from the language acquisition perspective, we might expect fundamental constraints that operate on structure building to induce learning biases. The two key constraints we consider here are *C(onstituent)-command* and *Locality*.

C-command is a relation that can be defined between constituents (i.e., nodes) that are merged. Adapting Reinhart's (1976) original definition to Minimalism:

(2) A node *A C-commands* a node *B iff*
i. *A* is merged with *X*, and
ii. *B* is merged within *X*

Considering the structure described in (1), *e* C-commands all other nodes, while *c* and *d* none. C-command is a fundamental property for various linguistic phenomena, such as agreement (3), gap licensing (4), and pronominal binding (5):

(3) [The friends [of John]] perform/*-s well.

(4) Joel discovered [the vase]$_i$ [that Patricia might take _$_i$]. / *Joel discovered [what$_i$ Patricia might take _$_i$ the vase].

(5) [A guy]$_i$ [that has seen [the wheelbarrow]$_j$] notices himself$_i$ /*itself$_j$.

Examples (4) and (5) are taken from BLiMP, but while (5) correctly illustrates our point—the referent *a guy* C-commands the anaphor *himself*, while *the wheelbarrow* does not C-command *itself*, despite being closer to the potential anaphor— the contrast illustrated by the minimal pair (4) is a bit misleading. The ungrammatical version in the minimal pair (4) is *Joel discovered what Patricia might take the vase*. This is an example of a "doubly filled gap": the gap position in (4) is not only filled with a DP—*the vase*—but it must also be interpreted as the legitimate argumental position in which the *wh-* item—*what*—should have been originally merged. Notice that in this contrast, no C-command violation arises. (3), on the other hand, perfectly illustrates that a closer DP *John* that does not C-command—{{the friends {of {John}}} perform}—the relevant predicate *perform* cannot agree with it.

Locality selectively restricts the span of a re-Merge (aka *Move*) operation (Rizzi, 2013). A straightforward example is illustrated by the intervention effects ("superiority effect", in the case of (6), Chomsky, 1973): a dependency between two nodes is blocked or disturbed by the presence of an intervening constituent, which is itself a potential participant in that dependency—e.g. it C-commands the gap, that is, the position where the relevant wh- item must be interpreted. Observe how ungrammaticality ensues when the *wh*-element *who* blocks the movement of the *wh*-element *what*, which is "moved" from/to its argumental position:

(6) a. *What$_i$* could Alan discover he has run around _$_i$?

b. **What* could Alan discover *who* has run around _$_i$?

More constraints. Other contrasts are illustrated in BLiMP that adhere to C-command and Locality but also involve additional considerations and constraints that we cannot address here. These considerations and constraints are relevant to *Ellipsis*, *Control*, and *Raising* phenomena, which, despite notable attempts to describe them under a unified account, remain empirically resistant to unification.

3.1 Computational Considerations on C-command and Locality

Our core idea was to modify the gating system of a RNN to allow the network to decide whether to merge items sequentially—{*a x*}, where *a* and *x* are two tokens processed in this order <*a*, *x*>—or to instantiate a nested constituent—{*a* {*x*}}. When processing an embedded constituent, the superordinate phrasal information must be retained in memory and preserved for further merge operations that might occur once the embedded constituent initialized by *x* will be completed—{*a* {*x* …} *y*}, where *y* is the next token merged with *a*, after the closure of the constituent *x*. Moreover, any item merged within a nested phrase should be "ignored" at the superordinate level, meaning that any relevant structural relations (e.g., agreement or gap licensing) in the higher phrase should fall outside the scope of the nested items.

The RNN architecture we adopted—loosely inspired by expectation-based Minimalist Grammar formalism (Chesi, 2022)—is dubbed eMG-RNN and implements two pathways, as in standard LSTMs: one for "movement"—non-local dependencies sensitive to locality and C-command, (the *c* output in the graph in Figure 3)—, the other for "Merge" that finally affects the output *h*.

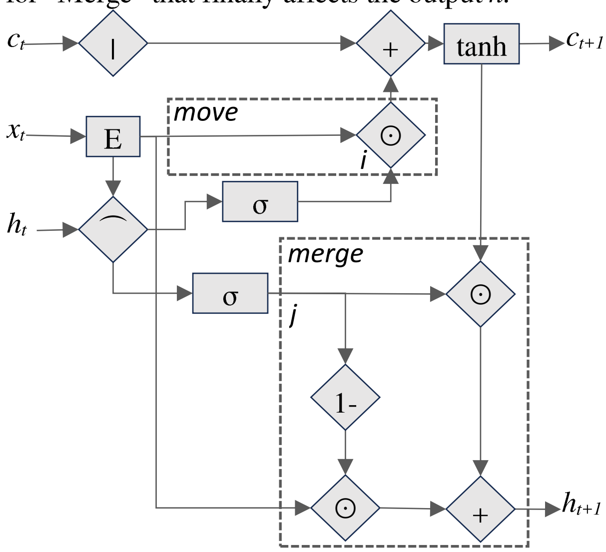


Figure 3: eMG-RNN computational graph

The gating system is slightly more complex compared to the one of LSTMs and GRUs: to contribute to the non-local activation *c*, the incoming token is first combined with the previous activation *h*, then transformed (*sigmoid*) before being applied (⊙) to the input to decide if any relevant dependency is fully satisfied or not (*move* gate); on the other pathway, the input and the previous output h are combined (*merge* gate). As in the update GRUs gate, if the *merge* gate activation is robust, the incoming input will be favored (*nesting* condition); on the other hand, smaller merge activation will favor a continuation with c activation (*sequential* processing). The non-local c-activation is further transformed (*tanh*) before being passed to the next layer/output, in order to stabilize the output and model short memory decay (Lewis and Vasishth, 2005).

More precisely:

(7)
$$move_t = \sigma(W_{xi}x_t \frown W_{hi}h_{t-1}) \odot W_{ii}x_t$$
$$merge_t = \sigma(W_{xj}x_t \frown W_{hj}h_{t-1})$$
$$c_{t+1} = tanh(c_t + move_t)$$
$$h_{t+1} = (1 - merge_t) \odot W_{xi}x_t + merge_t \odot c_{t+1}$$

As an anonymous reviewer observed, those modifications do not guarantee at all that what we are modeling here as *move* and *merge* gates are, in fact, the Minimalist Move and Merge operations. The simple gating mechanisms employed merely assume that these operations are *unification* processes (Shieber, 1986; Chesi, 2022), where extracted features are combined and the result of unification—whether after Merge or Move—is the outcome of a simple combination of the original vectors. We propose that point-wise multiplication ($\odot$) between the concatenation of hidden $\frown$ input vectors and the input vector itself, resulting from embedding (E) is the simplest way to test this intuition. In the following experiments, we selectively modify one component (*move gate*) or the other (*merge gate*) to verify whether the reduction in accuracy resulting from these alterations aligns with the linguistic predictions that motivated this gating system.

4 Methodology

To test the new gating system, we built various RNN architectures in PyTorch (v2.4). We first implemented very simple LSTM and GRU networks, similar to the ones discussed in literature (Gulordava et al., 2018; Warstadt et al., 2020; Chowdhury and Zamparelli, 2018). Input and hidden layer(s) normalization has been evaluated and produced a slight improvement in accuracy (+.02%) and a decrease in training loss (-.2) on average when compared to unnormalized layers. Various dropout options at the input level have been tested as well but removed to reduce training loss and increase accuracy—with a dropout=.2 the average cross entropy loss increased of .6 and a 11% accuracy reduction was observed, which is coherent with the small size of the networks used. We trained these architectures with the 10M corpus for a maximum of 20 epochs—all architecture reached a plateau at worst after 12 epochs. Both symmetrical—same number of units for the embedding layer and for the hidden layers, and asymmetrical structures—lower embedding dimensions, higher number of units in the hidden layers (Chowdhury and Zamparelli, 2018) have been tested. Following Lau et al. (2017), the model output is the negative sum of the token-by-token log likelihood—cross entropy loss—, normalized by the input length. All models used a BPE tokenizer (Sennrich et al., 2016) trained on the corpus with *min_freq* set to 3 to reduce lexicon size and speed-up training (no significant improvements are observed removing this frequency constraint). The lexicon obtained consisted of 67,328 tokens. For all experiments, the maximum sequence length was 74, batch size = 64 and learning rate = 0.002. We used torch.optim.lr_scheduler with step_size=5, gamma=0.1. We also used three different data batching regimens for training. We refer to the default maximum sequence length approach as the *redundant* regimen: the corpus was divided into overlapping sequences of 74 tokens each, disregarding sentence segmentation—[[$token_1$, $token_2$, … $token_{74}$], [$token_2$, $token_3$, … $token_{75}$], …]. This produces an exposure to ~740M tokens per epoch, which is about ten times the exposure received by 7 y.o. children. We also tested two alternative regimens, which we consider more ecological. The first is the *naturalistic* regimen, which involves line-by-line batching with no sentence segmentation special tokens, or overlapping—[[$token_1$, $token_2$, … $token_n$], [$token_{n+1}$, $token_{n+2}$, … $token_m$], …], resulting in an exposure to ~10M tokens per epoch. The second is the *conversational* regimen, where batches consist of two lines of variable length from the preprocessed text with no sentence segmentation tokens, but with one line/sentence overlapping to include minimal contextual information—[[$tokenized\_sentence_1$, $tokenized\_sentence_2$], [$tokenized\_sentence_2$, *tokenized_sentence$_3$*], …]. This doubles the exposure of the naturalistic regimen, while remaining within the order of magnitude of a 7-year-old's linguistic exposure. Training was performed on a High-Performance Cluster with 2 GPU nodes, each equipped with 64 CPU cores, 4 NVIDIA A100 cards with a dedicated 1GB RAM each. Each iteration required from ~1 (single-layer GRU) to ~20 hours (4-layer eMG-RNN).

4.1 Two ways of forgetting

One crucial experiment was to simplify the *move* and the *merge* gates to verify the effects of these simplifications both on training and BLiMP task performance. In the "forget nesting" condition (F-N), h_{t+1} became:

(8) $h_{t+1}=merge_t \odot c_{t+1}$

In the "forget moving" condition (F-M), the *move* gate became:

(9) $move_t=\sigma(W_{xi}x_t \frown W_{hi}h_{t-1})$

Our predictions are summarized below:

Prediction 1. If the gating system is sufficient to express C-command and Locality, all BLiMP pairs contrasting these aspects should be captured by eMG-RNN, but not by standard GRU or LSTM.

Prediction 2. Because of the sufficiently complex gating system, no improvement is expected building eMG-RNNs with multiple hidden layers.

Prediction 3. Selectively removing one gate or the other should affect performance; however, alteration of the *move* gate is expected to produce a more significant performance deterioration—this is because the simplification of the *nesting* mechanism will simply privilege sequential processing.

5 Results

Because of the low performance of GRUs (training results with 650 units for 1-, 2- or 3-layer respectively: accuracy=.3649, .3376, .3271, loss=3.1619, 3.3312, 3.4313; BLiMP supplement=.4410, .4426, .4390, filtered=.5162, .5161, .5362), we report here only the comparisons between eMG-RNNs and standard LSTMs.

Training performance. All architectures trained under the *naturalistic* and *conversational* regimen obtained low loss value (1.98 on average) and very high accuracy (90% on average) since the first epoch—plateau after 2-3 epochs. With the *redundant* regimen, more variegate results are obtained but all architectures reached ceiling performance after ten or twelve training epochs—see Figure 4 for the best performances with this last training regimen. Asymmetric architectures (E256_H1500) achieved better training performances (higher accuracy, lower loss$_{CE}$). This higher training performance is comparable with the one obtained with symmetric LSTM (E650, H650) when only the CHILDES section was used for training (child-directed speech only regimen). No significant differences have been found adding extra layers in both architectures (ceiling performance reached with 2-3 layers in LSTMs, with 1 layer in eMG-RNNs).

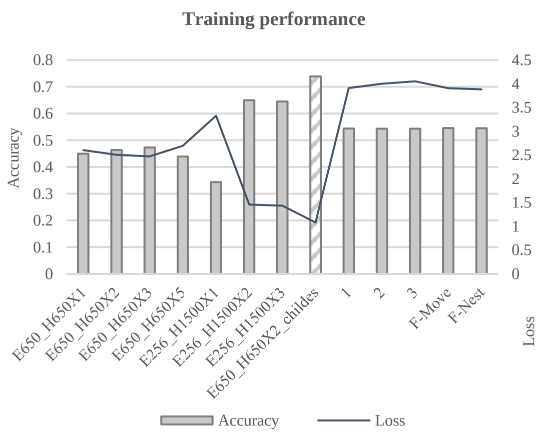


Figure 4: Best results under the redundant regimen with various LSTM architectures (labels represent architectures: E*m*_H*n*X*o* = *LSTM with m nodes as input, n nodes at the hidden layer and o layers*) eMG-RNN with *1*, *2* or *3* layers, eMG-RNN with deficient Move gate (*F-Move*) or nesting gate (*F-Nest*).

BLiMP accuracy. *Redundant* regimen produced the best results—*naturalistic* and *conversational* regimens induced a performance drop for the best architecture of ~10% and a dramatic reduction in consistency—calculated as in footnote 2. A significant lower performance is observed with LSTMs trained on 10M corpus with respect to the LSTM trained on 90M tokens Wikipedia corpus reported in the original study —Table 1. The cumulative results are reported in Table 3. The best performing LSTM architecture was the E650 H650 (Gulordava et al., 2018). Overall, the performance of this LSTM model trained only on the CHILDES section was not significantly different (overall performance on BLiMP supplement=0.47, filtered=0.54). All eMG-RNNs, outperform LSTMs on BLiMP filtered (0.55-0.59) but the performance on BLiMP supplement is lower (0.45-0.46). Even though the cumulative performance remains low, eMG-RNN models show much more polarized preferences and very low standard errors—Appendix A. That is, accuracy on *wh*-islands and other *wh*-dependency ranges from .80 to .96, while NPI licensing goes from .02 to .11, clearly indicating a preference for the ungrammatical sentence in the pair.

		eMG-RNN				
	LSTM	1	2	3	F-M	F-N
Ana. agr	0.67	0.82	0.76	0.77	0.88	0.81
Arg. str	0.56	0.65	0.64	0.63	0.64	0.66
Binding	0.54	0.69	0.66	0.63	0.57	0.65
Ctrl. / Rais.	0.59	0.58	0.59	0.60	0.58	0.60
D-N agr	0.57	0.67	0.63	0.67	0.68	0.68
Ellipsis	0.41	0.24	0.30	0.21	0.42	0.39
Filler. gap	0.55	0.64	0.60	0.47	0.48	0.65
Irregular	0.54	0.58	0.69	0.60	0.60	0.58
Island	0.54	0.58	0.54	0.53	0.50	0.62
Npi	0.45	0.33	0.50	0.55	0.32	0.31
Quantifiers	0.57	0.55	0.53	0.53	0.53	0.57
S-V agr	0.50	0.52	0.52	0.52	0.55	0.53
Overall	0.54	0.58	0.58	0.57	0.55	0.59

Table 3. Aggregated performance on BLiMP. LSTM is a 2 hidden-layer network (E650-H650), eMG-RNN networks are respectively 1, 2 and 3 layers, 1-layer simplified Move (F-M) and Merge/Nesting gate (F-N)

No significant difference is observed in performance adding extra layers to the eMG-RNN models, though eMG-RNN with three layers, perform randomly on NPIs, filler-gap dependencies and islands. As far as islands are concerned, it is important to notice that the aggregate results are little informative: while 1-leyer eMG-RNN performance is random on adjunct islands, it is systematically correct on wh-islands. Lastly, simplifying the Move gate produces a significant performance drop, while even better results are obtained by "forgetting" about nesting—Merge gate simplification.

6 Discussion

Once again (Feng et al., 2024), re-exploring RNN architectures produced some noteworthy outcomes. First, we observed that with simple architectures and limited training data, classic LSTMs and GRUs are insufficient to capture meaningful linguistic generalizations. On the other hand, adopting a different gating approach, designed to support structural biases during training, leads to an improvement in forced-choice linguistic tasks. While overall accuracy remains low, this average performance conceals the interesting fact that the eMG-RNN models consistently prefer (up to 44% of the phenomena vs. 0.04% with the best performant LSTM) one option over the other—Appendix A for details. Even when the chosen option is incorrect—as in the case of NPIs—this indicates that structural biases are operative. Furthermore, as evidenced by the low standard error, lexical perturbation is marginal compared to structural inference. This point is further supported by the very low accuracy on semantic tasks, such as those required when the BLiMP supplement is performed: eMG-RNNs produce insufficient semantic generalizations. Since the goal of these experiments was to explore the transparency of simple gating options in relation to certain relevant linguistic intuitions, we conclude that our attempt is partially successful even though the gating system must be improved to capture phenomena such as control, operator-variable licensing and ellipsis. Regarding the original predictions, our experiments confirm that: (i) the gating system adopted significantly outperforms both LSTM and GRU architectures in terms of structural inferences; (ii) additional hidden layers do not improve the models' performance on structural contrasts—these architectures exhibit a very low semantic/lexical bias; and (iii) the *Move* gate appears to be much more fundamental than nesting control. This result may be consistent with the fact that, in the proposed contrasts, nesting resolution is required only in a small number of cases—something we also tend to avoid in spoken language. Lastly, the *redundant* regimen is the only one that produces effective improvement on BLiMP tasks, independent of training performance. This confirms that, despite their cognitive plausibility, these architectures do not yet challenge the Poverty of Stimulus hypothesis.

Acknowledgments

This project is partially supported by the T-GRA2L: Testing GRAdeness and GRAmmaticality in Linguistics, PRIN 2022 Next Generation EU funded Project (202223PL4N). National coordinator: CC

References

Bever, T. G. 1970. The cognitive basis for linguistic structures. *Cognition and the development of language*.

Bianchi, V. and Chesi, C. 2014. Subject islands, reconstruction, and the flow of the computation. *LINGUISTIC INQUIRY*(4):525–569.

BNC Consortium. 2007. The British National Corpus, XML Edition.

Charpentier, L. G. G. and Samuel, D. 2023. Not all layers are equally as important: Every Layer Counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 210–224, Singapore. Association for Computational Linguistics.

Chesi, C. 2022. Expectation-based Minimalist Grammars: using the same root knowledge parsing and generation. *IJCOL*.

Chesi, C. and Canal, P. 2019. Person Features and Lexical Restrictions in Italian Clefts. *FRONTIERS IN PSYCHOLOGY*.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv:1406.1078 [cs, stat].

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*.volume 11. MIT Press, Cambridge, MA.

Chomsky, N. 1973. Conditions on transformations. In S. Anderson and P. Kiparsky, editors, *A Festschrift for Morris Halle*, pages 232–286. Holt, Rinehart and Winston, New York.

Noam Chomsky. 1981. *Lectures on government and binding: The Pisa lectures*. Walter de Gruyter.

Noam Chomsky, T. Daniel Seely, Robert C. Berwick, Sandiway Fong, M. A. C. Huybregts, Hisatsugu Kitahara, Andrew McInnerney, and Yushi Sugimoto. 2023. *Merge and the Strong Minimalist Thesis*. Cambridge University Press, 1st ed.

Chowdhury, S. A. and Zamparelli, R. 2018. RNN Simulations of Grammaticality Judgments on Long-distance Dependencies. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 133–144, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Elman, J. L. 1990. Finding Structure in Time. *Cognitive Science*, 14(2):179–211.

Ettinger, A. 2020. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Fallon, J. and Pylkkänen, L. 2024. Language at a glance: How our brains grasp linguistic structure from parallel visual input.

Feng, L., Tung, F., Ahmed, M. O., Bengio, Y., and Hajimirsadegh, H. 2024. Were RNNs All We Needed? arXiv:2410.01201 [cs].

Frazier, L. 1987. Syntactic Processing: Evidence from Dutch. *Natural Language & Linguistic Theory*, 5(4):519–559.

Gerlach, M. and Font-Clos, F. 2020. A Standardized Project Gutenberg Corpus for Statistical Analysis of Natural Language and Quantitative Linguistics. *Entropy*, 22(1):126.

Godfrey, J. J., Holliman, E. C., and McDaniel, J. 1992. SWITCHBOARD: Telephone speech corpus for research and development. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 1:517–520.

Gu, A. and Dao, T. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752 [cs].

Gulordava, K., Bojanowski, P., Grave, E., Linzen, T., and Baroni, M. 2018. Colorless Green Recurrent Networks Dream Hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kobele, G. M., Gerth, S., and Hale, J. 2013. Memory Resource Allocation in Top-Down Minimalist Parsing. In Glyn Morrill and Mark-Jan Nederhof, editors, *Formal Grammar*, volume 8036 of *Lecture Notes in*

Computer Science, pages 32–51. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kuncoro, A., Dyer, C., Hale, J., Yogatama, D., Clark, S., and Blunsom, P. 2018. LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

Lau, J. H., Clark, A., and Lappin, S. 2017. Grammaticality, Acceptability, and Probability: A Probabilistic View of Linguistic Knowledge. *Cognitive Science*, 41(5):1202–1241.

Lewis, R. L. and Vasishth, S. 2005. An activation-based model of sentence processing as skilled memory retrieval. *Cognitive science*, 29(3):375–419.

Lison, P. and Tiedemann, J. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles.

B MacWhinney. 2000. *The CHILDES Project: Tools for analyzing talk*. Mahwah, NJ: Lawrence Erlbaum Associates, Third edition.

Pham, T. M., Bui, T., Mai, L., and Nguyen, A. 2021. Out of Order: How Important Is The Sequential Order of Words in a Sentence in Natural Language Understanding Tasks?

Reinhart, T. 1976. *The syntactic domain of anaphora*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge (MA).

Rizzi, L. 2013. Locality. *Lingua*, 130:169–186.

Sennrich, R., Haddow, B., and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. arXiv:1508.07909 [cs].

Stuart M Shieber. 1986. *An introduction to unification-based approaches to grammar*.Lecture Notes. CSLI, Stanford, CA.

Snell, J. and Grainger, J. 2017. The sentence superiority effect revisited. *Cognition*, 168:217–221.

Stabler, E. 2013. Two Models of Minimalist, Incremental Syntactic Analysis. *Topics in Cognitive Science*, 5(3):611–633.

Steuer, J., Mosbach, M., and Klakow, D. 2023. Large GPT-like Models are Bad Babies: A Closer Look at the Relationship between Linguistic Competence and Psycholinguistic Measures. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 114–129, Singapore. Association for Computational Linguistics.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C., and Meeter, M. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. 2017. Attention Is All You Need. *arXiv:1706.03762 [cs]*. arXiv: 1706.03762.

Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., and Bowman, S. R. 2020. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Xu, W., Wu, Q., Liang, Z., Han, J., Ning, X., Shi, Y., Lin, W., and Zhang, Y. 2024. SLMRec: Empowering Small Language Models for Sequential Recommendation. arXiv:2405.17890 [cs].

Yang, C., Crain, S., Berwick, R. C., Chomsky, N., and Bolhuis, J. J. 2017. The growth of language: Universal Grammar, experience, and principles of computation. *Neuroscience & Biobehavioral Reviews*, 81:103–119.

Zaccarella, E. and Friederici, A. D. 2015. Merge in the Human Brain: A Sub-Region Based Functional Investigation in the Left Pars Opercularis. *Frontiers in Psychology*, 6.

A Appendice – Detailed BLiMP results

	LSTM		eMG-RNN							
	E650_H650X2		650x1		650x2		F-C		F-N	
categories	**acc**	**stderr**	**acc**	**stderr**	**acc**	**stderr**	**acc**	**stderr**	**acc**	**stderr**
BLiMP supplement	**0.56**	**0.01**	**0.46**	**0.01**	**0.45**	**0.01**	**0.45**	**0.01**	**0.47**	**0.01**
- hypernym	0.54	0.02	0.54	0.02	0.51	0.02	0.49	0.02	0.52	0.02
- qa congruence easy	0.42	0.06	0.33	0.06	0.31	0.06	0.39	0.06	0.36	0.06
- qa congruence tricky	0.50	0.04	0.31	0.04	0.34	0.04	0.30	0.04	0.30	0.04
- subject aux inversion	0.57	0.01	0.59	0.01	0.54	0.01	0.54	0.01	0.66	0.01
- turn taking	0.49	0.03	0.53	0.03	0.55	0.03	0.55	0.03	0.50	0.03
BLiMP filtered	**0.54**	**0.00**	**0.58**	**0.00**	**0.58**	**0.00**	**0.55**	**0.00**	**0.59**	**0.00**
- adjunct island filtered	0.45	0.02	0.49	0.02	0.35	0.02	0.43	0.02	0.51	0.02
- anaphor gender agreement filtered	0.68	0.01	0.77	0.01	0.70	0.01	0.82	0.01	0.80	0.01
- anaphor number agreement filtered	0.65	0.02	0.87	0.01	0.81	0.01	0.94	0.01	0.81	0.01
- animate subject passive filtered	0.58	0.02	0.58	0.02	0.59	0.02	0.62	0.02	0.60	0.02
- animate subject trans filtered	0.75	0.01	0.87	0.01	0.88	0.01	0.87	0.01	0.87	0.01
- causative filtered	0.45	0.02	0.54	0.02	0.52	0.02	0.50	0.02	0.59	0.02
- complex NP island filtered	0.45	0.02	0.50	0.02	0.42	0.02	0.60	0.02	0.55	0.02
- coordinate structure constraint complex left branch filtered	0.62	0.02	0.57	0.02	0.61	0.02	0.68	0.02	0.92	0.01
- coordinate structure constraint object extraction filtered	0.42	0.02	0.37	0.02	0.34	0.02	0.31	0.02	0.25	0.01
- determiner noun agreement 1 filtered	0.55	0.02	0.68	0.02	0.65	0.02	0.67	0.02	0.67	0.02
- determiner noun agreement 2 filtered	0.58	0.02	0.69	0.02	0.66	0.02	0.73	0.01	0.66	0.02
- determiner noun agreement irregular 1 filtered	0.57	0.02	0.64	0.02	0.60	0.02	0.60	0.02	0.69	0.02
- determiner noun agreement irregular 2 filtered	0.65	0.02	0.73	0.02	0.70	0.02	0.80	0.01	0.69	0.02
- determiner noun agreement with adj 2 filtered	0.54	0.02	0.64	0.02	0.58	0.02	0.66	0.02	0.61	0.02
- determiner noun agreement with adj irregular 1 filtered	0.55	0.02	0.62	0.02	0.57	0.02	0.65	0.02	0.77	0.02
- determiner noun agreement with adj irregular 2 filtered	0.56	0.02	0.70	0.02	0.67	0.02	0.75	0.01	0.68	0.02
- determiner noun agreement with adjective 1 filtered	0.55	0.02	0.65	0.02	0.60	0.02	0.60	0.02	0.64	0.02
- distractor agreement relational noun filtered	0.47	0.02	0.48	0.02	0.51	0.02	0.46	0.02	0.47	0.02
- distractor agreement relative clause filtered	0.49	0.02	0.51	0.02	0.51	0.02	0.48	0.02	0.50	0.02
- drop argument filtered	0.60	0.02	0.75	0.01	0.72	0.01	0.71	0.01	0.72	0.01
- ellipsis n bar 1 filtered	0.49	0.02	0.27	0.02	0.34	0.02	0.54	0.02	0.51	0.02
- ellipsis n bar 2 filtered	0.33	0.02	0.21	0.01	0.26	0.02	0.30	0.02	0.28	0.02
- existential there object raising filtered	0.65	0.02	0.67	0.02	0.65	0.02	0.63	0.02	0.72	0.02
- existential there quantifiers 1 filtered	0.63	0.02	0.94	0.01	0.90	0.01	0.90	0.01	0.97	0.01
- existential there quantifiers 2 filtered	0.81	0.01	0.30	0.02	0.06	0.01	0.57	0.02	0.43	0.02
- existential there subject raising filtered	0.62	0.02	0.56	0.02	0.66	0.02	0.66	0.02	0.60	0.02
- expletive it object raising filtered	0.62	0.02	0.58	0.02	0.56	0.02	0.57	0.02	0.57	0.02
- inchoative filtered	0.41	0.02	0.42	0.02	0.40	0.02	0.47	0.02	0.43	0.02
- intransitive filtered	0.42	0.02	0.62	0.02	0.62	0.02	0.64	0.02	0.65	0.02
- irregular past participle adjectives filtered	0.61	0.02	0.68	0.02	0.76	0.01	0.53	0.02	0.77	0.01
- irregular past participle verbs filtered	0.46	0.02	0.49	0.02	0.63	0.02	0.68	0.02	0.38	0.02
- irregular plural subject verb agreement 1 filtered	0.52	0.02	0.59	0.02	0.54	0.02	0.64	0.02	0.61	0.02
- irregular plural subject verb agreement 2 filtered	0.51	0.02	0.57	0.02	0.58	0.02	0.59	0.02	0.57	0.02
- left branch island echo question filtered	0.82	0.01	0.46	0.02	0.33	0.02	0.42	0.02	0.46	0.02
- left branch island simple question filtered	0.57	0.02	0.66	0.02	0.69	0.02	0.63	0.02	0.89	0.01
- matrix question npi licensor present filtered	0.24	0.01	0.48	0.02	0.69	0.02	0.77	0.01	0.07	0.01
- npi present 1 filtered	0.41	0.02	0.11	0.01	0.27	0.01	0.13	0.01	0.25	0.01
- npi present 2 filtered	0.41	0.02	0.10	0.01	0.26	0.01	0.12	0.01	0.23	0.01
- only npi licensor present filtered	0.58	0.02	0.37	0.02	0.71	0.02	0.00	0.00	0.72	0.02
- only npi scope filtered	0.36	0.02	0.02	0.01	0.01	0.00	0.07	0.01	0.31	0.02
- passive 1 filtered	0.62	0.02	0.75	0.02	0.75	0.01	0.72	0.02	0.76	0.01
- passive 2 filtered	0.66	0.02	0.76	0.01	0.75	0.01	0.70	0.02	0.75	0.01
- principle A c command filtered	0.41	0.02	0.63	0.02	0.54	0.02	0.62	0.02	0.54	0.02
- principle A case 1 filtered	0.73	0.01	1.00	0.00	1.00	0.00	0.59	0.02	0.92	0.01
- principle A case 2 filtered	0.50	0.02	0.62	0.02	0.63	0.02	0.60	0.02	0.72	0.01
- principle A domain 1 filtered	0.53	0.02	0.69	0.02	0.49	0.02	0.45	0.02	0.67	0.02
- principle A domain 2 filtered	0.53	0.02	0.50	0.02	0.55	0.02	0.51	0.02	0.47	0.02
- principle A domain 3 filtered	0.54	0.02	0.54	0.02	0.55	0.02	0.51	0.02	0.54	0.02
- principle A reconstruction filtered	0.53	0.02	0.88	0.01	0.88	0.01	0.72	0.01	0.73	0.01
- regular plural subject verb agreement 1 filtered	0.49	0.02	0.53	0.02	0.52	0.02	0.68	0.02	0.57	0.02
- regular plural subject verb agreement 2 filtered	0.49	0.02	0.44	0.02	0.47	0.02	0.47	0.02	0.49	0.02
- sentential negation npi licensor present filtered	0.62	0.02	0.66	0.02	0.70	0.02	0.57	0.02	0.38	0.02
- sentential negation npi scope filtered	0.54	0.02	0.60	0.02	0.84	0.01	0.61	0.02	0.25	0.01
- sentential subject island filtered	0.43	0.02	0.70	0.01	0.71	0.01	0.55	0.02	0.58	0.02
- superlative quantifiers 1 filtered	0.36	0.02	0.51	0.02	0.51	0.02	0.51	0.02	0.51	0.02
- superlative quantifiers 2 filtered	0.47	0.02	0.47	0.02	0.64	0.02	0.15	0.01	0.37	0.02
- tough vs raising 1 filtered	0.37	0.02	0.37	0.02	0.30	0.01	0.36	0.02	0.39	0.02
- tough vs raising 2 filtered	0.67	0.02	0.69	0.02	0.76	0.01	0.67	0.02	0.69	0.02
- transitive filtered	0.59	0.02	0.54	0.02	0.55	0.02	0.52	0.02	0.55	0.02
- wh island filtered	0.58	0.02	0.89	0.01	0.86	0.01	0.42	0.02	0.83	0.01
- wh questions object gap filtered	0.61	0.02	0.80	0.01	0.77	0.01	0.38	0.02	0.85	0.01
- wh questions subject gap filtered	0.63	0.02	0.82	0.01	0.80	0.01	0.20	0.01	0.81	0.01
- wh questions subject gap long distance filtered	0.68	0.02	0.82	0.01	0.69	0.02	0.60	0.02	0.89	0.01
- wh vs that no gap filtered	0.59	0.02	0.95	0.01	0.88	0.01	0.55	0.02	0.97	0.01
- wh vs that no gap long distance filtered	0.65	0.02	0.96	0.01	0.85	0.01	0.59	0.02	0.97	0.01
- wh vs that with gap filtered	0.38	0.02	0.07	0.01	0.08	0.01	0.58	0.02	0.03	0.01
- wh vs that with gap long distance filtered	0.35	0.02	0.07	0.01	0.16	0.01	0.44	0.02	0.02	0.00

Developmentally Plausible Multimodal Language Models Are Highly Modular

Alina Klerings[1] **Christian Bartelt**[1] **Aaron Mueller**[2,3]
[1] University of Mannheim [2] Northeastern University [3] Technion – IIT
alina.klerings@uni-mannheim.de aa.mueller@northeastern.edu

Abstract

Large language models demonstrate emergent modularity, where functionally specialized components and circuits arise to handle specific tasks or task formats. If similar modules arise in models trained on more cognitively plausible datasets, it could inform debates surrounding what kinds of mechanisms would be learnable given more human-like language learning signals. In this paper, we describe a multimodal vision-language model submitted to the BabyLM Challenge. Our model achieves similar performance to the best-performing architectures from last year, though visual information does not improve performance on text-only tasks over text-only models (in accordance with prior findings). To better understand how the model processes the evaluation tasks of the BabyLM Challenge, we leverage causal interpretability methods to locate the neurons that contribute to the model's final decisions. We find that the models we train are highly modular: distinct components arise to process related tasks. Furthermore, on text-and-image tasks, adding or removing visual inputs causes the model to use distinct components to process the same textual inputs. This suggests that modal and task-specific specialization is efficiently learned, and that a high degree of functional specialization arises in even small-scale language models.

1 Introduction

Despite impressive capabilities across a wide range of tasks, language models (LMs) remain highly data-inefficient: LMs typically require orders of magnitude more data during pretraining than humans encounter over their entire lifetime (Gilkerson et al., 2017). This inefficiency has driven interest in alternative approaches to language learning that leverage more human-like language learning scenarios. One such effort is the BabyLM Challenge (Warstadt et al., 2023), which promotes the development of language models trained on the quantity of linguistic input that children receive when learning language. To create a more developmentally plausible training setup, the 2024 iteration of the challenge (Choshen et al., 2024) provides aligned image and text data.

Evaluating these more cognitively plausible models requires a focused analysis not only of how models behave, but also of the mechanisms[1] underlying their behaviors. Conventional benchmarks are finite and often deploy identically distributed train/test splits, causing us to overlook key aspects of how models generalize. To address this, mechanistic interpretability has emerged as a framework for obtaining a more algorithmic understanding of how neural networks perform particular behaviors. This typically entails causally attributing model behavior to specific components, or causal graphs composed thereof.

We conduct a study around one of the baseline architectures from the BabyLM Challenge that incorporates both language and vision: the generative image transformer (GIT; Wang et al., 2022). We train and evaluate a suite of language-only and multimodal models with this architecture to investigate the role of visual inputs in language learning. Specifically, we first examine how different weighting schemes for text and image-text loss signals affect model performance and assess whether visual input offers any benefit for language learning. As expected, visual data leads to enhanced performance on multimodal benchmarks compared to text-only models. However, we find no significant benefit of visual data for performance on text-only benchmarks. This supports prior findings of a multimodal submission from last year's BabyLM Challenge (Amariucai and Warstadt, 2023), as well as findings of Zhuang et al. (2024).

Then, using attribution patching (Syed et al., 2023), we identify the most causally important neu-

[1]At a high level, a mechanism can be defined as a causal graph describing how inputs are transformed into outputs.

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 118–139
November 15-16, 2024

rons in GIT's text decoder across tasks. This analysis reveals a high level of modularity,[2] with separate internal mechanisms being deployed even for slightly different subtasks of the same task. Most surprisingly, the same textual input is processed differently in the text decoder depending on whether visual inputs are present. This suggests that visual inputs do not merely add to textual information, but rather activate distinct mechanisms in the model's language processing components. These findings suggest that modal and task-specific specialization is efficiently learnable in human-like learning scenarios, even in the absence of human-like learning biases.[3] These findings extend prior work on emergent modularity in pre-trained language models (e.g., Zhang et al., 2023; Csordás et al., 2021; Agarwala et al., 2021) to a more cognitively plausible training scenario, thus allowing us to make more convincing claims as to what kinds of linguistic functional specializations can arise from human-like language learning signals.

Our main contributions are as follows:

- An analysis of what small-scale language models gain from visual inputs over pure text.
- A causal analysis of which text decoder neurons perform each BabyLM evaluation task, and how the addition of vision data changes these component sets.
- A suite of minimally differing autoregressive text-only and text-and-image models for future analyses.[4]

2 Related Work

Small-scale multimodal language modeling Many believe that grounding text data in some symbolic representation or alternate modality is necessary for robust language understanding (Bender and Koller, 2020; Bisk et al., 2020, *inter alia*). Thus, assuming the training corpus is no more than what a human could realistically be exposed to when learning language, the addition of aligned visual data may provide an even better test ground for understanding what kinds of structures are learnable from data alone (without a human-like inductive bias). Recent related work has investigated whether visual inputs can aid in word learning, finding largely negative results—but crucially, visual inputs *are* helpful in the kinds of low-resource scenarios we investigate (Zhuang et al., 2024). The 2023 BabyLM Challenge received many multimodal submissions; most relevant to ours is the text-and-vision submission of Amariucai and Warstadt (2023).

Mechanistic interpretability Mechanistic interpretability methods allow us to more deeply understand where and how particular tasks are accomplished in a neural network. This paper focuses more on *localizing* than qualitatively *explaining* model behavior—but localization can itself reveal whether certain behaviors are performed using the same underlying mechanisms. For example, one line of work aims to causally quantify whether the most important neurons for a particular task overlap with those from highly related tasks in language models (e.g., Finlayson et al., 2021; Sankaranarayanan et al., 2024). There also exist investigations of the mechanisms underlying how vision-language models accomplish particular tasks (e.g., Palit et al., 2023; Salin et al., 2022). Past work has used other (not always causal) methods to discover that language models are highly modular; this includes work with small-scale CNN and LSTM-based models (Csordás et al., 2021; Agarwala et al., 2021), as well as large Transformer-based models (Zhang et al., 2023).

Our work extends this literature through analyses of developmentally plausible multimodal language models. We investigate whether these models use similar mechanisms to perform diverse natural language processing (NLP) tasks, and whether they use the same mechanisms to perform the same tasks with and without image data. While our models are not directly comparable to human learners due to differing inductive biases and a relatively small quantity of visual inputs, they nonetheless provide evidence as to the kinds of mechanisms that are learnable from a realistic language learning dataset.

3 Methods

3.1 Model Training

We closely replicate the challenge baseline setup as a foundation for our causal analysis, with the goal of mechanistic insights rather than model optimization. Specifically, we train a series of generative image transformer (GIT; Wang et al., 2022)

[2] In this context, "modularity" refers to function-based neuron grouping (Zhang et al., 2023), where particular neuron clusters have specific functions.

[3] This degree of modularity is not necessarily desirable nor undesirable; see §5.

[4] Our code and models are publicly available: https://github.com/klerings/babylm_analysis

models on the official training data for the multimodal track of the BabyLM Challenge (Choshen et al., 2024). The corpus is composed of two parts: one half consists of text-only data—primarily transcribed speech and child-directed language—while the other half is composed of paired image-caption data from sources such as Localized Narratives (Pont-Tuset et al., 2020) and Conceptual Captions (Sharma et al., 2018).

GIT Architecture The GIT architecture consists of two main components: an image encoder and a text decoder. For the image encoder, we use DINOv2 (Oquab et al., 2024), a Vision Transformer (ViT; Dosovitskiy et al., 2021), which is pretrained independently in a self-supervised manner using only image data, thus not counting towards the word budget imposed by the challenge. The text decoder is then jointly pretrained with the image encoder on image-text pairs, following a causal language modeling objective.

GIT also offers the advantage that it can function as a decoder-only language model when image input is absent, enabling additional training on text-only data and facilitating evaluation on both unimodal and multimodal tasks.

Multimodal Loss GIT uses a standard cross-entropy loss for language modeling, which is computed over two types of training data: (1) samples containing both images and text (from Localized Narratives and Conceptual Captions) and (2) text-only samples (from the BabyLM corpus). These two types of data are handled separately during training, with distinct loss terms for each.

For samples that include both images and text, the model computes a loss by predicting the caption tokens, conditioned on the preceding text tokens and the projected image encoding. This loss is denoted as $\mathcal{L}_{\text{multi}}$. Notably, the image input from this corpus can be disabled to simulate a language-only model.

For text-only samples (from the BabyLM corpus), the model computes a unimodal loss, $\mathcal{L}_{\text{uni}}$, where each token is predicted based solely on the preceding text tokens.

The total loss during training is a weighted sum of these two components:

$$\mathcal{L} = w_1 \mathcal{L}_{\text{multi}} + w_2 \mathcal{L}_{\text{uni}} \tag{1}$$

We investigate the impact of varying weight configurations[5]. A configuration denoted as 1/1 implies equal weighting ($w_1 = w_2$), while 1/0.5 refers to $w_1 = 1$ and $w_2 = 0.5$.

When we include images in the captions corpus, the weights w_1 and w_2 not only determine the degree of emphasis placed on child-directed language in the BabyLM corpus, but also adjust the contributions of multimodal and unimodal loss signals during training. For more information on implementation and hyperparameters, see App. A.

3.2 Benchmarks

We evaluate our models on the official benchmarks of the BabyLM Challenge to verify their competitiveness with the challenge baselines and ensure relevance of any conclusions drawn from the subsequent analysis. For language understanding this includes BLiMP (Warstadt et al., 2020), BLiMP Supplement (Warstadt et al., 2023), EWoK (Ivanova et al., 2024) and GLUE (Wang et al., 2018, 2020), see Table 7 in App. C.2 for examples. BLiMP and its supplement consist of sentence pairs with one grammatically correct and one incorrect sentence. EWoK tests logical entailment requiring world knowledge and reasoning, where the model must choose the more semantically likely of two continuations given prior context. Accuracy on BLiMP and EWoK is measured by how often the model assigns a higher probability to the correct sentence. Meanwhile, GLUE tests natural language understanding after task-specific finetuning.

To assess combined textual and visual understanding, the BabyLM Challenge evaluates on the visual question answering benchmark VQAv2 (Goyal et al., 2019) using 7 distractor answers, as well as on Winoground (Thrush et al., 2022) and DevBench (Tan et al., 2024). Winoground includes images paired with two sentences: one accurately describing the image, and another minimally differing sentence that reflects a contrasting scenario. For samples in DevBench, the model must instead select one of multiple images given a textual concept or scenario. These are each evaluated in a zero-shot manner. In addition to the BabyLM evaluation tasks, we evaluate on the visual question answering benchmark MMStar (Chen et al., 2024), which has been manually curated to exclude questions that could be answered via linguistic information alone.

[5]Since increasing the relative importance of one loss component is equivalent to decreasing the importance of the other, we only experiment with varying w_1.

3.3 Baselines

We compare our baseline replication against the released baselines from the BabyLM competition. For text-and-vision tasks, this includes Flamingo (Alayrac et al., 2024) and GIT, which are trained on the multimodal BabyLM training corpus. For text-only tasks, this also includes last year's winning architectures, BabyLlama (Timiryasov and Tastet, 2023) and LTG-BERT (Georges Gabriel Charpentier and Samuel, 2023), both trained on the official training data from the Strict track, comprising the same number of words as the multimodal corpus.[6]

3.4 Attribution Patching

We causally attribute model behaviors to specific neurons to determine whether the most important components are shared across task settings. A key technique for this purpose is **attribution patching** (Syed et al., 2023) with integrated gradients (AP-IG; (Hanna et al., 2024; Marks et al., 2024)), a linear approximation of the computationally more expensive activation patching (Vig et al., 2020; Finlayson et al., 2021; Geiger et al., 2021). Activation patching entails intervening on the activation of a model component during a forward pass; the extent to which this intervention changes the model behavior is measured as the **indirect effect** (IE). Activation patching is often used with contrastive input pairs, where activations from one prompt are transferred into a forward pass on a minimally different prompt. It also supports interventions like setting the activation to zero[7] or replacing the activation with its mean across some dataset.

In attribution patching, rather than directly patching neuron activations, the indirect effect is linearly approximated by multiplying the gradient of the target metric m with respect to the neuron's activation x by the difference between the original activation x and the counterfactual activation x':

$$\hat{\text{IE}} = \frac{\delta m}{\delta x} \cdot (x' - x) \tag{2}$$

The gradient can be viewed as a local approximation of how much changing the neuron's activation would affect m, so multiplying this by how much x changes gives us an estimate of how much m will change. Typically, m is the logit difference between a correct token completion and minimally different incorrect token completion. High-magnitude $\hat{\text{IE}}$ values indicate that a neuron significantly influences a particular model behavior.[8]

Benchmark-specific prompts and metrics For BLiMP, we select a subset of subtasks consistent with the "one-prefix-method" (Linzen et al., 2016) which ensures that both sentences of a pair share an initial phrase but diverge at a critical word that determines grammaticality. This format generalizes well to VQA, where the logit difference is computed between the target answer and the first distractor that consists of a single token.

Attribution patching is primarily suited to cases where the correct and counterfactual answers can be distinguished by a single token. This is not the case for the other tasks of the BabyLM challenge. Therefore, we adapt the prompt structure and target metric to suit the specific nature of each benchmark, as illustrated in Table 7 in App. C.2.

While MMStar has a multiple-choice structure similar to VQA, the answer choices often exceed a single token in length, rendering the single-token logit difference metric unsuitable. For EWoK and Winoground, the tasks are not formulated as question-answer pairs; instead, the objective is to select the more plausible sentence given a preceding sentence or image. Accordingly, we employ an alternative metric that compares the sum of logits for the entire correct sentence S_1 against the sum for the entire incorrect sentence S_2, given a textual or visual context. In other words, $m = \sum_{s_1 \in S_1} p(s_1) - \sum_{s_2 \in S_2} p(s_2)$. For EWoK, we repeat the context sentence following the first context and continuation; these are separated by a newline, allowing the model to process the full text input for each comparison (and thus allowing us to backpropagate after comparing $p(s_1)$ and $p(s_2)$). In Winoground, the context consists of the image representation, and both possible description sentences separated by newlines. Similarly, for MMStar, the prompt is made up of the image and both question-answer pairs (where the question is repeated), separated by newlines.

This design presents a challenge: prior work has shown that language models can be semantically

[6]But from a different distribution. The 50M words of image-caption data are replaced by data more closely resembling the text-only corpus's distribution.

[7]This is not entirely principled and may even be out-of-distribution for the network, as a neuron's baseline value will not necessarily be 0.

[8]This includes positive as well as negative $\hat{\text{IE}}$ values. An example of components that *negatively* and significantly impact performance are Negative Name Mover Heads in the Indirect Object Identification task (Wang et al., 2023).

	BLiMP	BLiMP-Supp.	EWoK	GLUE	*Avg.*	*Avg. w/o GLUE*
			Baseline Models			
BabyLlama (100M)	**73.1**	60.6	52.1	69.0	63.7	61.9
LTG-BERT (100M)	69.2	**66.5**	51.9	68.4	64.0	62.5
Flamingo	70.9	65.0	**52.7**	**69.5**	**64.5**	62.9
GIT	65.2	62.7	52.4	68.3	62.2	**65.1**
			Multimodal Models			
GIT 1/1	70.0 (2.03)	65.8 (2.26)	51.9 (0.75)	-	-	62.6
GIT 1/0.5	68.9 (1.41)	64.1 (1.96)	**52.7** (0.40)	-	-	61.9
GIT 1/0.25	71.2 (1.34)	64.6 (2.29)	52.5 (0.20)	-	-	62.8
GIT 1/0.125	66.3 (1.88)	61.7 (1.44)	52.3 (0.91)	65.6	61.5	60.1
			Language-only Models			
GIT 1/1	72.0 (1.54)	65.6 (1.89)	51.9 (0.39)	66.5	64.0	63.2
GIT 1/0.25	71.6 (1.22)	64.0 (2.32)	52.6 (0.38)	-	-	62.7

Table 1: Results for text-only benchmarks averaged across 3 random seeds. Avg. columns refer to macroaverage over the respective tasks. For GIT, we show the corpus weightings as w_1/w_2.

and syntactically primed (Meyer and Schvaneveldt, 1971; Neely, 1977; Bock, 1986) to favor text more similar to prior text that has already been seen in the same context (van Schijndel and Linzen, 2018; Prasad et al., 2019). Therefore, we randomly alternate the order of correct and incorrect continuations to account for priming effects on average across examples. While this will not yield accurate *behaviors* per se, we care more about the relative probability *change* between $p(S_1)$ and $p(S_2)$ when a component is ablated, rather than their actual values; this design will still allow us to measure this quantity when averaging across inputs.

For each benchmark, we retrieve the 100 most important MLP neurons in the text decoder by $\hat{\mathrm{IE}}$ over all layers. We obtain the top neurons for each subtask within a benchmark. For some tasks that do not have subtasks such as VQA and Winoground, we automatically generate subcategorizations of examples. For more information on subtask definitions and the automatic subcategorization procedure, see App. C.1. We exclude DevBench from this analysis because its samples consist of multiple images, each requiring a separate forward pass, rendering attribution patching unfeasible.

4 Results

We train and evaluate four weighting configurations for the multimodal model and two for the text-only model; for each configuration, we average across three random seeds. Detailed information on the learning progress of each model is provided in App. B.

4.1 Benchmarking Results

We use the challenge benchmarks to validate that our models perform sufficiently well for meaningful neuron analysis. To explore the impact of visual information on language-only and multimodal learning, we evaluate all models on both text-only[9,10] and text-vision benchmarks.

Furthermore, we test the multimodal model's performance on vision tasks without image input, simulating its behavior as a language-only model. The average and standard deviation across all random seeds are presented in Tables 1 and 2.

Text-only Results For the text-only benchmarks, our models are on par with or slightly below the performance of the baseline models, except for GLUE, which we exclude from our causal attribution study.

There is no single weighting configuration that consistently performs best across all datasets, but

[9]Due to computational constraints, only the best model per modality and random seed is reported for GLUE. The best unimodal model is selected from 1/1 to ensure a fair comparison with other language-only models that similarly balance loss signals across all samples.

[10]The GLUE metric is an unweighted mean of each subtask accuracy, except QQP and MRPC (where we use F1 scores), and CoLA (where we use the Matthews correlation coefficient).

Input	VQA multimodal	VQA text-only	Winoground multimodal	Winoground text-only	DevBench multimodal	MMStar multimodal	MMStar text-only	*Avg.* multimodal
Baseline Models								
Flamingo	52.3	45.0	51.6	50.0	**60.1**	24.1	22.6	**47.0**
GIT	**54.1**	48.4	55.5	50.0	50.5	25.9	22.4	46.5
Multimodal Models								
GIT 1/1	51.5 (3.52)	49.2 (1.01)	55.4 (0.13)	50.0 (0.0)	48.7 (1.22)	25.1 (0.35)	23.0 (0.57)	45.1
GIT 1/0.5	53.1 (1.40)	47.5 (1.09)	55.9 (2.46)	50.0 (0.0)	50.2 (1.50)	24.3 (0.57)	21.8 (0.98)	45.7
GIT 1/0.25	52.2 (1.12)	47.4 (0.81)	56.2 (0.79)	50.0 (0.0)	47.6 (0.75)	25.8 (0.18)	22.5 (0.73)	45.3
GIT 1/0.125	52.6 (1.40)	48.6 (0.68)	**57.0** (0.66)	50.0 (0.0)	47.8 (2.52)	**26.7** (0.52)	22.6 (1.41)	45.9
Language-only Models								
GIT 1/0.1	-	49.4 (0.72)	-	50.0 (0.0)	-	-	22.9 (1.33)	-
GIT 1/0.25	-	48.0 (0.60)	-	50.0 (0.0)	-	-	24.0 (1.21)	-

Table 2: Results for multimodal benchmarks with (multimodal) and without (text-only) visual input averaged across 3 random seeds. "Avg." is a macroaverage over multimodal tasks. For GIT, we show loss weightings as w_1/w_2.

the models achieving the highest average performance are 1/1 for the language-only setup and 1/0.25 in the multimodal case. This is contrary to observations regarding the evaluation loss (App. B), where lower weightings on BabyLM data samples (w_2) correlated with performance improvement.

No significant performance differences are observed between models trained on textual data alone and those incorporating both text and image inputs, when comparing the same weightings. This suggests that the addition of multimodal data does not yield measurable improvements in this specific context. This aligns with findings from Zhuang et al. (2024).

Multimodal Results In multimodal tasks, our models exceed baseline performance on Winoground and MMStar but show a slight underperformance on VQA and a more significant drop on DevBench.

Results from both language-only and multimodal models without visual input provide validation and confirm that performance decreases substantially when image inputs are excluded. As on the text-only tasks, there is no single multimodal weighting configuration that consistently outperforms across all benchmarks. However, for tasks such as Winoground and MMStar, which require visual input for an above chance performance, the 1/0.125 weighting configuration proves most effective, as it places significantly more emphasis on the visual loss signal during training.

We present learning curves for the best-performing models in each modality across the BabyLM evaluation tasks in Figure 5 in App. B. For the multimodal model, we observe an order in which phenomena are acquired: BLiMP performance peaks early, whereas EWoK performance gradually improves later in training. In App. B, we discuss this order of acquisition further, and discuss how learning curves differ between multimodal and text-only models.

4.2 Causal Neuron Analysis

To explore whether neuron activation patterns are shared across tasks or modalities, we compute the average indirect effect for each MLP neuron in the text decoder of the strongest multimodal GIT model (1/0.125) per subtask. Then, we select the top 100 neurons by indirect effect and analyze their overlap across subtasks from all benchmarks.

Modularity within benchmarks For text-only benchmarks, the results (Figure 1) indicate a significant degree of neuron sharing in GIT among subtasks within each benchmark. Specifically, for EWoK, over 70% of the top neurons are pairwise shared between subtasks. However, given the low performance on the EWoK benchmark, it is possible that these neurons are not responsible for task solving, but rather pick up on spurious heuristics; we therefore focus on BLiMP and VQA[11]. Here, we observe a similar though less pronounced trend of intra-benchmark neuron sharing. For BLiMP,

[11]Note that VQA questions have seven distractor answers, so random chance performance is 12.5%

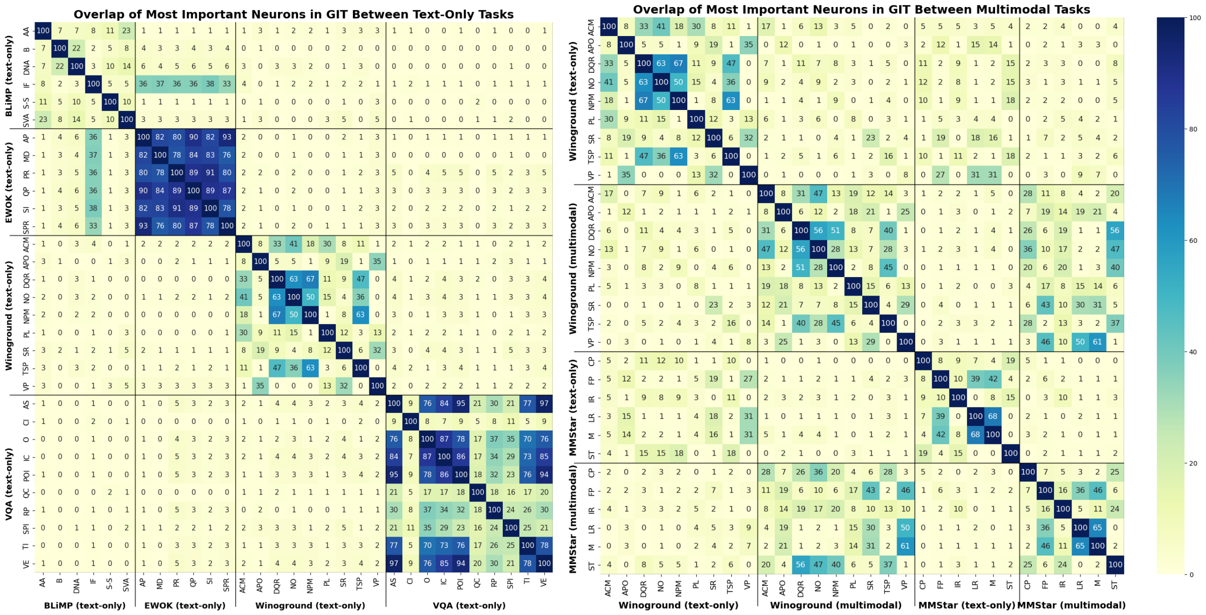


Figure 1: Overlap between top 100 neurons by $\hat{\mathrm{IE}}$ per subtask for text-only (*left*) and multimodal benchmarks with and without visual inputs (*right*). Subtask names are abbreviated; see App. C.1 for full names and example counts.

there is an overlap of about 20% between two subtask pairs. In VQA, many subtask pairs even share over 70% of their task-relevant neurons.

Component sharing across benchmarks The primary factor in determining neuron overlap appears to be task similarity: subtasks within the same benchmark are more similar and display a stronger overlap, whereas tasks across different benchmarks are very distinct and share little neurons. The 30% overlap between Irregular Forms (IF) in BLiMP and all EWoK subtasks is an exception, but models did not score well on these tasks; these could therefore be encoding spurious heuristics or irrelevant information.

Distinct processing of multimodal input A shift in neuron overlap is observed when comparing the same subtasks with and without visual input (Figure 1; full results in Figure 6 in App. C.3). The addition of vision leads to greater overlap of important neurons between all pairs of tasks: for example, the overlap between subtasks in MMStar and Winoground is 30% or less without images, but rises to 40-60% for certain subtasks when visual input is introduced.[12] This increase in shared components is also observed between VQA and Winoground, as well as between VQA and MMStar. Interestingly, this increase in shared top components does not extend to intra-benchmark subtasks. Here, we find a mixture of subtask pairs that increase their overlap, mostly in VQA, and subtask pairs that decrease their overlap as in Winoground.

Furthermore, we find the overlap between the same task with and without vision to be minimal for both Winoground and MMStar. This suggests that the presence of visual input significantly changes the mechanisms employed by the language decoder to solve these tasks.

Neuron Sharing in Flamingo To evaluate how well our findings generalize to other multimodal architectures, we conduct a causal neuron analysis on the BabyLM Flamingo baseline model. Unlike GIT, which relies on self-attention, Flamingo integrates vision and text using cross-attention between a frozen image encoder and text decoder.

Flamingo exhibits a similar degree of intra- and inter-task neuron overlap as GIT, with overlap increasing when visual input is added (Figure 7 in App. C.3). However, in contrast to GIT, EWoK displays only selective subtask overlap, aligning more closely with patterns observed in other datasets.

Notably, there is a significant amount of shared neurons between text-only and image-text variants of VQA. This was not observed with GIT (see Figure 2). While adding image inputs in other multimodal cases alters the salient features in the text, Flamingo's processing of VQA suggests the image

[12] Note that this is the overlap between MLP neurons in the *text decoder*, not in the image encoder. It is not necessarily intuitive that adding visual information should change the text processing mechanisms to this degree.

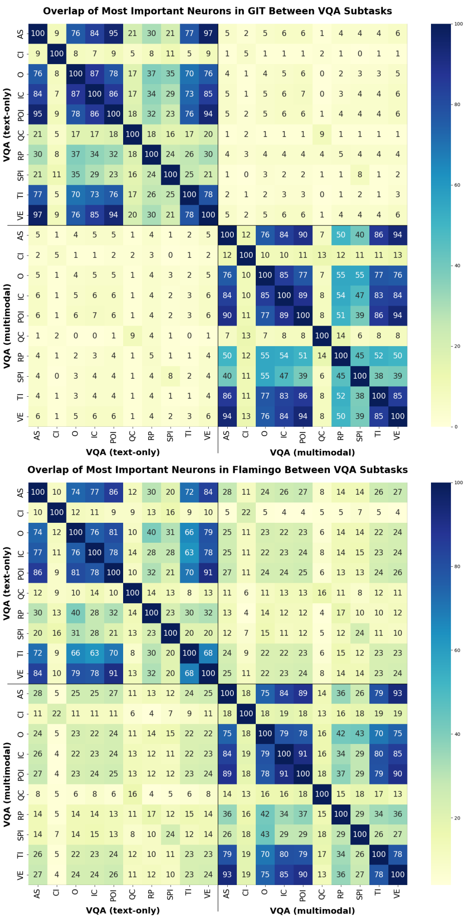


Figure 2: Overlap between top 100 neurons of GIT (*top*) and Flamingo (*bottom*) for subtasks of VQA with and without visual inputs.

supports rather than *redirects* the text decoder.

4.3 Neuron Ablation

To verify the causal influence of the identified top components, we perform a neuron ablation study on VQA. We mean ablate the most influential neurons for each subtask and measure the resulting performance changes, quantifying the effect of the removed information. We consider the top neurons of two settings: (1) text-only, where the multimodal GIT model processes just text, and (2) multimodal, integrating both text and visual inputs. We then mean ablate these distinct neuron sets in the multimodal model. We measure accuracy by the sign of the logit difference between correct answer and first distractor of token length one.

We witness an expected drop in GIT's performance for eight of the ten VQA subtasks (Figure 3; see Figure 8 in App. C.4 for all subtasks), confirming the task-relevance of the identified top neurons.

When measuring performance after ablations, we note four patterns. (1) Performance sometimes drops comparably when ablating only text-only neurons, or only text-image neurons. This could indicate that there are more task-relevant neurons shared than the overlap matrix of top 100 neurons implies, or simply that these two sets redundantly encode similar mechanisms. (2) Ablating text-image neurons sometimes results in a greater drop in performance. This suggests that the most important neurons are the ones processing the task multimodally, which could be indicative of successful fusion of vision and text data. (3) Some tasks experience a larger performance drop when ablating the text-only neurons, which means for these tasks, much of the model's performance can be attributed to question-answer likelihoods rather than visual reasoning. (4) There are two cases where performance *increases* after ablations: Color Identification and Quantity & Counting. Our models achieve comparatively low accuracies on these tasks before ablations; it is thus unclear whether these ablations improve scores because (i) the top neurons encode actively unhelpful spurious information, or (ii) ablating them causes the model to rely on some other heuristic that happens to be more successful (or both).

Similarly, in the pretrained Flamingo model, seven out of ten VQA subtasks show a performance decrease when either text-only or text-image neurons are ablated (Figure 9 in App. C.4). However, the drop is relatively small, indicating the model's robustness to MLP neuron ablations. This suggests that either more than 100 neurons are involved in task-relevant processes, or that critical processing takes place in other components of the architecture, such as the cross-attention mechanism.

5 Discussion

We find little neuron overlap between vision-and-text and text-only variants of the same tasks. This suggests a significant degree of modularity in small-scale multimodal language models.[13] This raises important questions: is component sharing between unimodal and multimodal processing mechanisms of the same task desirable? Can it serve as a signal of effective merging of information across modali-

[13] However, Flamingo's processing of VQA is an exception. This may be due to the training pipeline: text and image encoders are first trained separately, and then cross-attention between these frozen modules is learned using multimodal data. This contrasts with GIT, where text decoder and text-image associations are jointly learned.

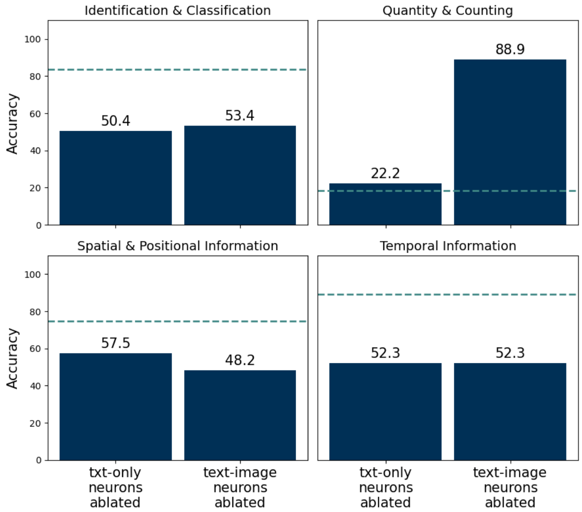


Figure 3: Multimodal accuracy for VQA subtasks when mean ablating GIT's top neurons. The dashed line indicates accuracy before ablations. The left and right bars show model performance given text and vision inputs when ablating either the top neurons from the text-only version of the task, or the top neurons from the text-and-image version of the task.

ties? To investigate, future research could explore the relationship between neuron overlap and task performance, ideally across diverse architectures.

Many causes could explain the minimal overlap across similar benchmarks. First, the conceptual space in the model representations could be such that there are few features or skills in common across tasks; thus, to the model, these tasks have little in common. Investigating this possibility would involve a more thorough qualitative analysis of the features implicated in performing each task. Second, different tasks may share features, but the model might learn domain-specific versions of qualitatively identical features. Follow-up research could vary task formats—for example, by paraphrasing all examples—and analyze whether this changes the top neurons. That said, there is some correlation between task similarity and component overlap within a benchmark. This serves as a sanity check, and also indicates that even small models tend to share processing mechanisms across closely related tasks with similar formats. This is a more parameter-efficient strategy compared to representing similar tasks in a fully modular fashion.

Is a high degree of task modularity desirable? Some argue that emergent modularity can be harnessed for better generalization in language models (e.g., Qiu et al., 2024); it could also enable more fine-grained mechanistic understanding and control. However, modularity will generally result in reduced parameter-efficiency. It could also be a signal that a model is not efficiently compressing information in a generalizable way, such that it must relearn similar phenomena for distinct task settings. We speculate that there exist more or less desirable types and extents of modularity in neural language models, and that classifying these types of modularity could be especially helpful in assessing parameter-(in)efficiency.

Relatedly, when speaking of modularity, it is essential to distinguish between two types of neural modules: (i) skill-related neural groups that share general abilities *independent of specific tasks*, and (ii) task-related neural groups that are specialized for particular *task formats*. In our experiments, we predominantly observe the latter. From an engineering perspective, there is no clear indication whether this would enhance performance or efficiency. However, if one's goal is to model human language processing, perhaps modularity could be a useful signal. Certain regions of the brain specialize toward particular tasks, even in the presence of similar visual stimuli across tasks (Dupont et al., 1993); different specialized regions for the same task can also arise given sufficiently distinct stimuli (Müller et al., 2024). Our findings agree with both. Whether emergent task modules in developmentally plausible language models correspond to comparable regions in the human brain remains an interesting open question.

6 Conclusion

Developmentally plausible multimodal language models exhibit a high degree of modularity. Furthermore, adding visual inputs changes how the text decoder processes a task, and increases the amount of shared components between tasks. Our findings highlight the types of functional specialization that can arise in language models trained on developmentally plausible data, and raise questions about trade-offs between sample-efficiency, parameter-efficiency, and cognitive plausibility.

Acknowledgments

This research is supported by the Ministry of Economic Affairs, Labor and Tourism of Baden-Württemberg and the bwHPC resources of Baden-Württemberg. A.M. is supported by a postdoctoral fellowship under the Zuckerman STEM Leadership Program.

Limitations

Our study focuses on two multimodal architectures. Other models such as CLIP combine visual and language data differently, and therefore, the influence of image data on the model's behaviors and mechanisms may be qualitatively different. Despite this, our current findings suggest that visual information does not significantly aid in language learning, highlighting the need for novel fusion strategies between the two modalities.

Additionally, there is room for improvement in the scope of the analyzed components during attribution patching. While we primarily examined MLP neurons, which are crucial for language generation, the role of attention layers impacts a model's decoding ability equally. Future work could investigate the influence of visual data on the emergence of task-specific attention heads, building on prior studies in mechanistic interpretability.

References

Atish Agarwala, Abhimanyu Das, Brendan Juba, Rina Panigrahy, Vatsal Sharan, Xin Wang, and Qiuyi Zhang. 2021. One network fits all? modular versus monolithic task formulations in neural networks. In *International Conference on Learning Representations*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millicah, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2024. Flamingo: a visual language model for few-shot learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Theodor Amariucai and Alexander Scott Warstadt. 2023. Acquiring linguistic knowledge from multimodal input. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 128–141, Singapore. Association for Computational Linguistics.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020. Experience grounds language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8718–8735, Online. Association for Computational Linguistics.

J.Kathryn Bock. 1986. Syntactic persistence in language production. *Cognitive Psychology*, 18(3):355–387.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. 2024. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

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.

Róbert Csordás, Sjoerd van Steenkiste, and Jürgen Schmidhuber. 2021. Are neural nets modular? inspecting functional modularity through differentiable weight masks. In *International Conference on Learning Representations*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Patrick Dupont, Guy A. Orban, Rufin Vogels, Guy Bormans, Johan Nuyts, Christiaan Schiepers, Michael De Roo, and Luc Mortelmans. 1993. Different perceptual tasks performed with the same visual stimulus attribute activate different regions of the human brain: A positron emission tomography study. *Proceedings of the National Academy of Sciences of the United States of America*, 90(23):10927–10931.

Matthew Finlayson, Aaron Mueller, Sebastian Gehrmann, Stuart Shieber, Tal Linzen, and Yonatan Belinkov. 2021. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1828–1843, Online. Association for Computational Linguistics.

Jaden Fiotto-Kaufman, Alexander R Loftus, Eric Todd, Jannik Brinkmann, Caden Juang, Koyena Pal, Can Rager, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Michael Ripa, Adam

Belfki, Nikhil Prakash, Sumeet Multani, Carla Brodley, Arjun Guha, Jonathan Bell, Byron Wallace, and David Bau. 2024. NNsight and NDIF: Democratizing access to foundation model internals.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 9574–9586. Curran Associates, Inc.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.

Jill Gilkerson, Jeffrey A Richards, Steven F Warren, Judith K Montgomery, Charles R Greenwood, D Kimbrough Oller, John HL Hansen, and Terrance D Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American journal of speech-language pathology*, 26(2):248–265.

Lila Gleitman. 1990. The structural sources of verb meanings. *Language acquisition*, 1(1):3–55.

Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2019. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *Int. J. Comput. Vision*, 127(4):398–414.

Jane Grimshaw. 1979. Complement selection and the lexicon. *Linguistic Inquiry*, 10(2):279–326.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms.

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*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 28(6):517–540.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models.

David E Meyer and Roger W Schvaneveldt. 1971. Facilitation in recognizing pairs of words: evidence of a dependence between retrieval operations. *Journal of experimental psychology*, 90(2):227.

Veronika I. Müller, Edna C. Cieslik, Linda Ficco, Sandra Tyralla, Amir Ali Sepehry, Taraneh Aziz-Safaie, Chunliang Feng, Simon B. Eickhoff, and Robert Langner. 2024. Not all stroop-type tasks are alike: Assessing the impact of stimulus material, task design, and cognitive demand via meta-analyses across neuroimaging studies. *Neuropsychology Review*.

James H Neely. 1977. Semantic priming and retrieval from lexical memory: Roles of inhibitionless spreading activation and limited-capacity attention. *Journal of experimental psychology: general*, 106(3):226.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. Dinov2: Learning robust visual features without supervision.

V. Palit, R. Pandey, A. Arora, and P. Liang. 2023. Towards vision-language mechanistic interpretability: A causal tracing tool for blip. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2848–2853, Los Alamitos, CA, USA. IEEE Computer Society.

Steven Pinker. 1984. *Language Learning and Language Development*. Harvard University Press.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting vision and language with localized narratives. In *Computer Vision – ECCV 2020*, pages 647–664, Cham. Springer International Publishing.

Grusha Prasad, Marten van Schijndel, and Tal Linzen. 2019. Using priming to uncover the organization of syntactic representations in neural language models. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Zihan Qiu, Zeyu Huang, and Jie Fu. 2024. Unlocking emergent modularity in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2638–2660, Mexico City, Mexico. Association for Computational Linguistics.

Emmanuelle Salin, Badreddine Farah, Stéphane Ayache, and Benoit Favre. 2022. Are vision-language transformers learning multimodal representations? a probing perspective. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11248–11257.

Aruna Sankaranarayanan, Dylan Hadfield-Menell, and Aaron Mueller. 2024. Disjoint processing mechanisms of hierarchical and linear grammars in large language models. In *ICML 2024 Workshop on LLMs and Cognition*.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia. Association for Computational Linguistics.

Aaquib Syed, Can Rager, and Arthur Conmy. 2023. Attribution patching outperforms automated circuit discovery. In *NeurIPS Workshop on Attributing Model Behavior at Scale*.

Alvin Wei Ming Tan, Sunny Yu, Bria Long, Wanjing Anya Ma, Tonya Murray, Rebecca D. Silverman, Jason D. Yeatman, and Michael C. Frank. 2024. Devbench: A multimodal developmental benchmark for language learning.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5238–5248.

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. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

Marten van Schijndel and Tal Linzen. 2018. A neural model of adaptation in reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4704–4710, Brussels, Belgium. Association for Computational Linguistics.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. Superglue: A stickier benchmark for general-purpose language understanding systems.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022. Git: A generative image-to-text transformer for vision and language.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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.

Zhengyan Zhang, Zhiyuan Zeng, Yankai Lin, Chaojun Xiao, Xiaozhi Wang, Xu Han, Zhiyuan Liu, Ruobing Xie, Maosong Sun, and Jie Zhou. 2023. Emergent modularity in pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4066–4083, Toronto, Canada. Association for Computational Linguistics.

Chengxu Zhuang, Evelina Fedorenko, and Jacob Andreas. 2024. Visual grounding helps learn word meanings in low-data regimes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1311–1329, Mexico City, Mexico. Association for Computational Linguistics.

A Model Training

A.1 Hyperparameters

We train all models for a maximum of 30 epochs, using a learning rate of 1e-4 with a weight decay

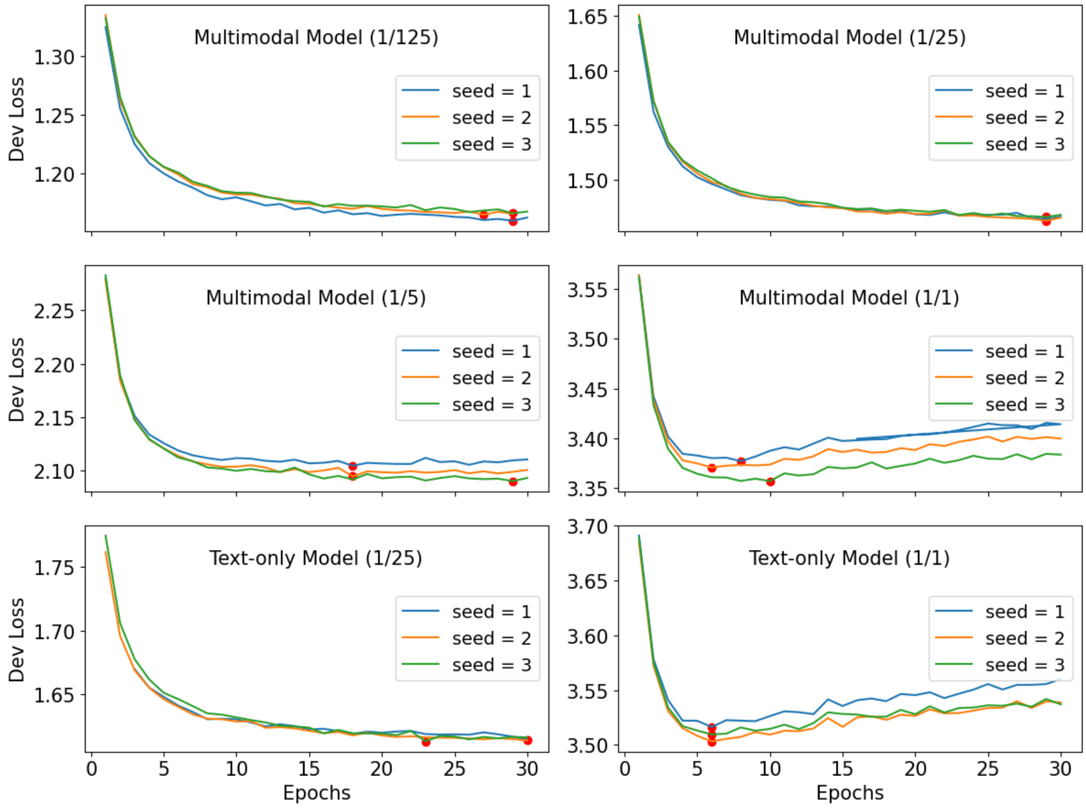


Figure 4: Evaluation loss of GIT per epoch for each of the weighting configurations across three random seeds. Red dot marks the best epoch.

of 0.1. The AdamW optimizer is employed with a batch size of 128, and early stopping is applied to prevent overfitting.

A.2 Tokenization

A single tokenizer is utilized across both unimodal and multimodal models to enhance comparability between the different settings. The tokenizer is trained on the BabyLM corpus as well as the image captions from the Localized Narratives and Conceptual Captions datasets, with a vocabulary size of 32,778 tokens.

B Learning Curves

After every epoch, we compute the validation loss on the unimodal or multimodal development set from the BabyLM Challenge, depending on the model we are working with. We provide learning curves for all weighting configuration in Figure 4.

In the 1/0.125 and 1/0.25 weighting configurations, the loss consistently decreases across seeds and modalities, indicating potential for further improvement with additional epochs. In contrast, the 1/0.5 multimodal models show convergence within the 30-epoch limit. For the 1/1 configuration, where train losses are evenly weighted, overfitting occurs after six to ten epochs in both unimodal and multimodal setups. We conclude that the fusion of language and vision is only learned reliably with a strong multimodal loss signal. For the language-only model, the setting with lower w_2 value exhibits the better convergence, suggesting that language skills and decoding abilities may be more effectively learned from non-child-directed language present in image captions.

We also present learning curves for each benchmarking task. Learning curves for the best-performing models in each modality across benchmarks are visualized in Figure 5. For the multimodal model, there appears to be an order in which phenomena are acquired: task performance on vision benchmarks and the EWoK dataset increases steadily. In contrast, performance on the BLiMP and BLiMP Supplement datasets peaks early in training and subsequently fluctuates or declines. We discuss this in more detail below. The language-only model shows minimal performance change over time on BLiMP, VQA, Winoground, and MMStar benchmarks, with performance remaining at initial levels. For the EWoK dataset, performance

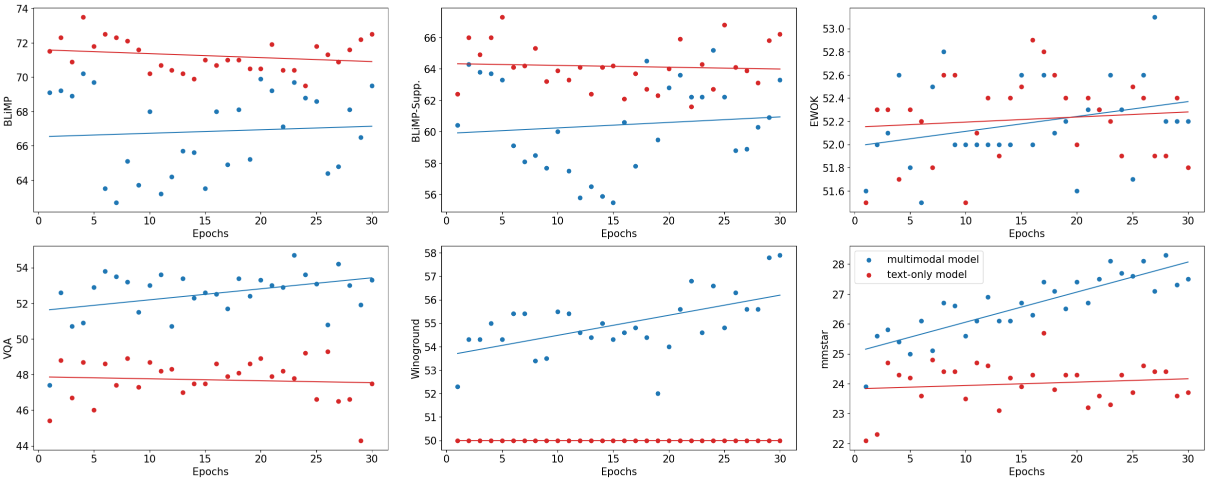


Figure 5: Learning progress of best GIT model per modality on each benchmark. Moving average smoothing is applied with window size 3.

peaks around 15 epochs before declining, whereas on the smaller BLiMP Supplement task, performance fluctuation occurs almost immediately.

These findings align with observations from evaluation loss curves, where the 1/0.125 multimodal model exhibits continued learning, while the 1/1 language-only model reaches an early local minimum.

Acquiring linguistic abilities in order. We observe a distinct order of acquisition in language models: learning curves across benchmarks indicate an almost immediate proficiency in distinguishing between valid and invalid formal linguistic structures, primarily with respect to morphosyntactic rules. This is reflected in the high scores achieved on both BLiMP and BLiMP Supplement early in training. In contrast, performance on EWoK, a benchmark that assesses more functional (semantic and pragmatic) linguistic abilities in context, improves gradually and slowly over time—and peaks at significantly lower scores. This phased "order of acquisition" deviates somewhat from human language development, where syntactic and semantic signals can assist in learning the other throughout language acquisition (Gleitman, 1990; Grimshaw, 1979; Pinker, 1984). This finding could support the existence of a clear distinction between effective representations of the formal structure of language, and representations of how language should be interpreted and deployed in context (Mahowald et al., 2024); nonetheless, this finding is preliminary and should be investigated in more depth and in a greater variety of architectures and learning scenarios.

C Causal Neuron Analysis

C.1 Subtask Categories

The subtask categories are either provided explicitly in the dataset (e.g. EWoK, BLiMP, MMStar) or automatically aggregated using a large language model. Since the subtask labels in VQA and Winoground are too fine-grained, we leverage ChatGPT-4o to automatically merge them to broader categories. This is achieved in a two-stage process, where we first ask for the generation of superclass labels and then for the assignment of these labels to the fine-grained categories. This process is done in two steps to ensure that each fine-grained label is assigned exactly one superclass, see Table 3

Label Creation
The following is a list of VQA/Winoground question types. It is too fine-grained, merge the categories to 10 combined categories that are reasonable to group together, and give the merged categories a new name...

Assignment
Classify each of these following question types with exactly one of these super categories...

Table 3: ChatGPT-4o prompts used to generate new subtask labels.

In Tables 4 and 5 we provide a mapping between original and superclass label per benchmark and in Table 6 we report the number of samples per supercategory, alongside an abbreviation used in heatmap plots.

Person and Object Identification			
are these	are they	is he	is it
is the man	is the person	is the woman	is this
is this person	what is the man	what is the person	what is the woman
what is this	who is		
Other (General Queries and Miscellaneous)			
are the	what does the	what is	do you
what is the	how	none of the above	what
what are	what are the		
Action and State			
can you	has	could	
Color Identification			
what color	what color is the	what color are the	what color is
what is the color of the			
Verification and Existence			
are	does the	is the	are there
does this	is there a	are there any	is
is there	do	is there	
Identification and Classification			
is that a	what animal is	what is the name	is this a
what kind of	what sport is	is this an	what type of
which	what brand		
Temporal Information			
was	what time		
Spatial and Positional Information			
how many people are in	what room is	where are the	what is in the
where is the	what is on the		
Reason and Purpose			
why	why is the		
Quantity and Counting			
how many	what number is	how many people are	

Table 4: VQA subtask categories with their original question types.

Adjectival Comparisons and Modifications		
Adjective-Age	Adjective-Size	Adjective-Manner
Adjective-Color	Adjective-Color (3-way swap)	Adjective-Shape
Adjective-Texture	Adjective-Animate	Adjective-Weight
Adjective-Temperature	Adjective-Speed	Adjective-Height
Adjective-Manner Phrase	Adjective-Speed Phrase, Verb-Intransitive	Adverb-Animate
Verb Phrases (Intransitive and Transitive)		
Verb-Intransitive	Verb-Transitive	Verb-Transitive Phrase, Verb-Intransitive, Preposition Phrase
Verb-Transitive Phrase	Verb-Intransitive, Noun	Verb-Intransitive Phrase
Verb-Intransitive, Determiner-Numeral	Verb-Intransitive, Adjective-Manner	Verb-Intransitive, Verb-Transitive Phrase
Verb-Intransitive Phrase, Adverb-Animate	Verb-Intransitive Phrase, Preposition	Verb-Transitive, Noun
Noun Phrases and Modifiers		
Noun, Adjective-Color	Noun Phrase, Adjective-Animate	Noun
Noun Phrase	Noun Phrase, Adjective-Color	Noun Phrase, Determiner-Possessive
Noun Phrase, Determiner-Numeral	Noun, Verb-Intransitive	Noun, Preposition Phrase, Scope
Noun, Adjective-Size		
Altered POS		
Sentence	Altered POS	Altered POS, Determiner-Numeral
Preposition and Locations		
Preposition Phrase, Scope	Preposition Phrase	Preposition
Determiner and Quantifier Relationships		
Determiner-Numeral	Determiner-Possessive	Determiner-Numeral Phrase
Determiner-Numeral, Noun Phrase		
Scope and Relations		
Scope	Scope, Preposition, Verb-Intransitive	Scope, Preposition Phrase
Scope, Adjective-Manner	Scope, Adjective-Texture	Scope, Conjunction Phrase
Scope, Relative Clause	Scope, Conjunction	Scope, Verb-Transitive
Scope, Preposition	Relative Clause, Scope	Scope, Preposition Phrase, Adjective-Color
Scope, Altered POS, Verb-Intransitive, Verb-Transitive	Scope, Noun, Preposition	
Negation and Opposites		
Negation, Scope	Negation, Noun Phrase, Preposition Phrase	
Temporal and Spatial Phrases		
Adjective-Temporal	Adjective-Spatial	Adverb-Temporal
Adverb-Spatial Phrase	Adverb-Spatial	

Table 5: Winoground subtask categories with their original question types.

BLIMP (linguistics_term)

Subtask Name	Abb.	Num.
Subject Verb Agreement	SVA	34
S-Selection	S-S	417
Anaphor Agreement	AA	688
Binding	B	1056
Determiner Noun Agreement	DNA	1710
Irregular Forms	IF	67

MMStar (category)

Subtask Name	Abb.	Num.
Fine-grained Perception	FP	247
Instance Reasoning	IR	243
Science and Technology	ST	174
Coarse Perception	CP	245
Math	M	112
Logical Reasoning	LR	204

VQA

Subtask Name	Abb.	Num.
Person and Object Identification	POI	3208
General Queries and Miscellaneous (renamed: Other)	O	4648
Action and State	AS	286
Color Identification	CI	2343
Verification and Existence	VE	4894
Identification and Classification	IC	2393
Temporal Information	TI	176
Spatial and Positional Information	SPI	708
Reason and Purpose	RP	100
Quantity and Counting	QC	27

Winoground

Subtask Name	Abb.	Num.
Adjectival Comparisons and Modifications	ACM	184
Verb Phrases (Intransitive and Transitive)	VP	52
Noun Phrases and Modifiers	NPM	268
Altered POS	APO	46
Preposition and Locations	PL	68
Determiner and Quantifier Relationships	DQR	50
Scope and Relations	SR	42
Negation and Opposites	NO	18
Temporal and Spatial Phrases	TSP	12

EWoK (Domain)

Subtask Name	Abb.	Num.
Physical Relations	PR	818
Spatial Relations	SPR	476
Physical Interactions	PI	556
Agent Properties	AP	2056
Material Dynamics	MD	770
Social Properties	SP	325
Social Relations	SOR	1548
Quantitative Properties	QP	310
Social Interactions	SI	294
Physical Dynamics	PD	120
Material Properties	MP	170

Table 6: (Aggregated) subtask categories per benchmark with their abbreviation and number of contained samples.

C.2 Prompt Format and Metrics

An example for each prompting format used in attribution patching is given in Table 7, alongside the metric used to compute the patching effect.

C.3 Heatmap for all subtasks

We provide an extensive heatmap for the neuron overlap between subtasks of all benchmarks in Figure 6 for GIT and in Figure 7 for Flamingo.

C.4 Neuron Ablation

We provide the ablation effect for all subtasks of VQA (in their multimodal variant) when ablating the top neurons with their mean activation in Figure 8 for GIT and in Figure 9 for Flamingo.

C.5 Library

To perform attribution patching and neuron ablations, we use `nnsight` (Fiotto-Kaufman et al., 2024).

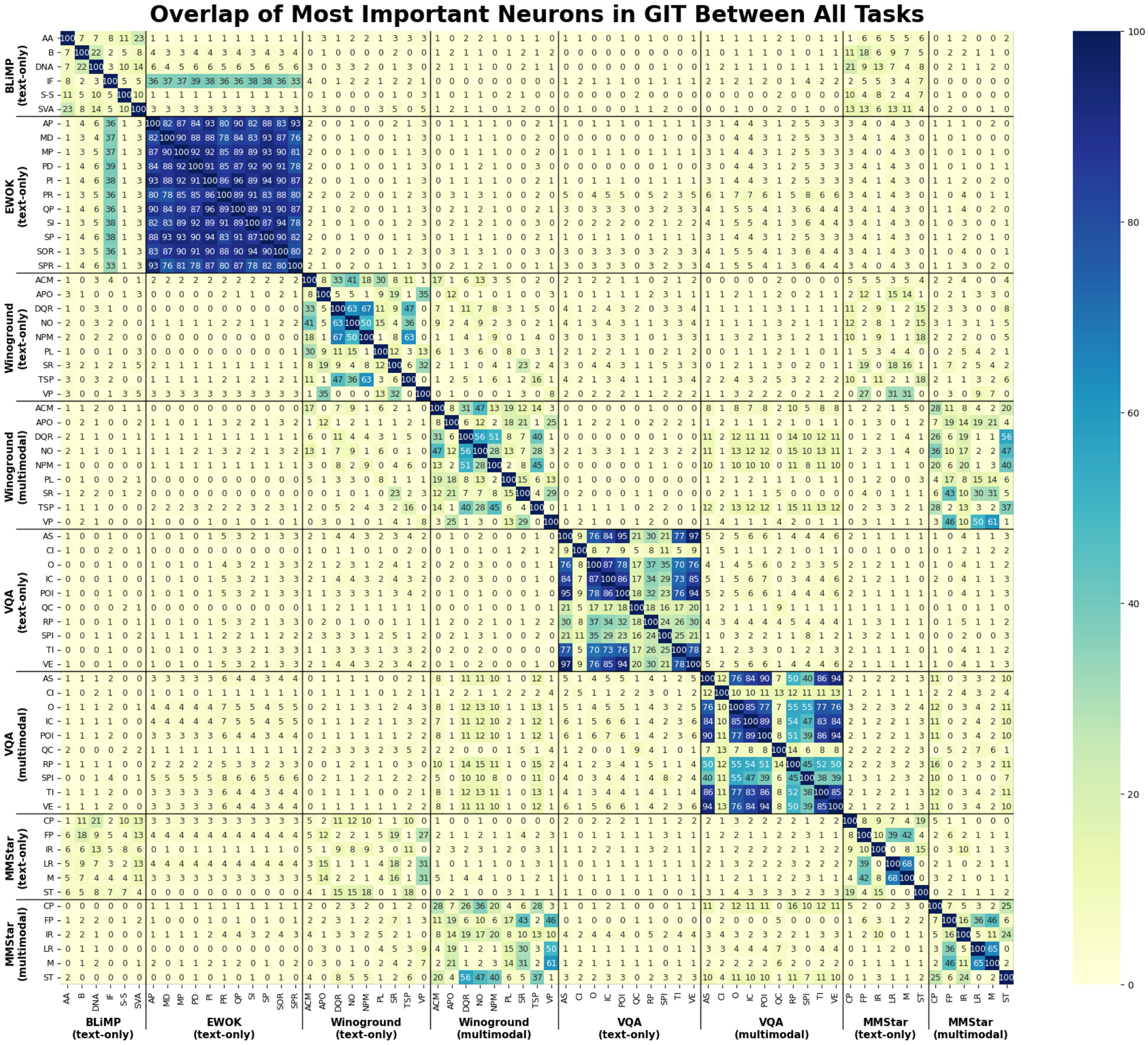


Figure 6: Overlap in GIT between the top 100 neurons by indirect effect per subtask for all benchmarks.

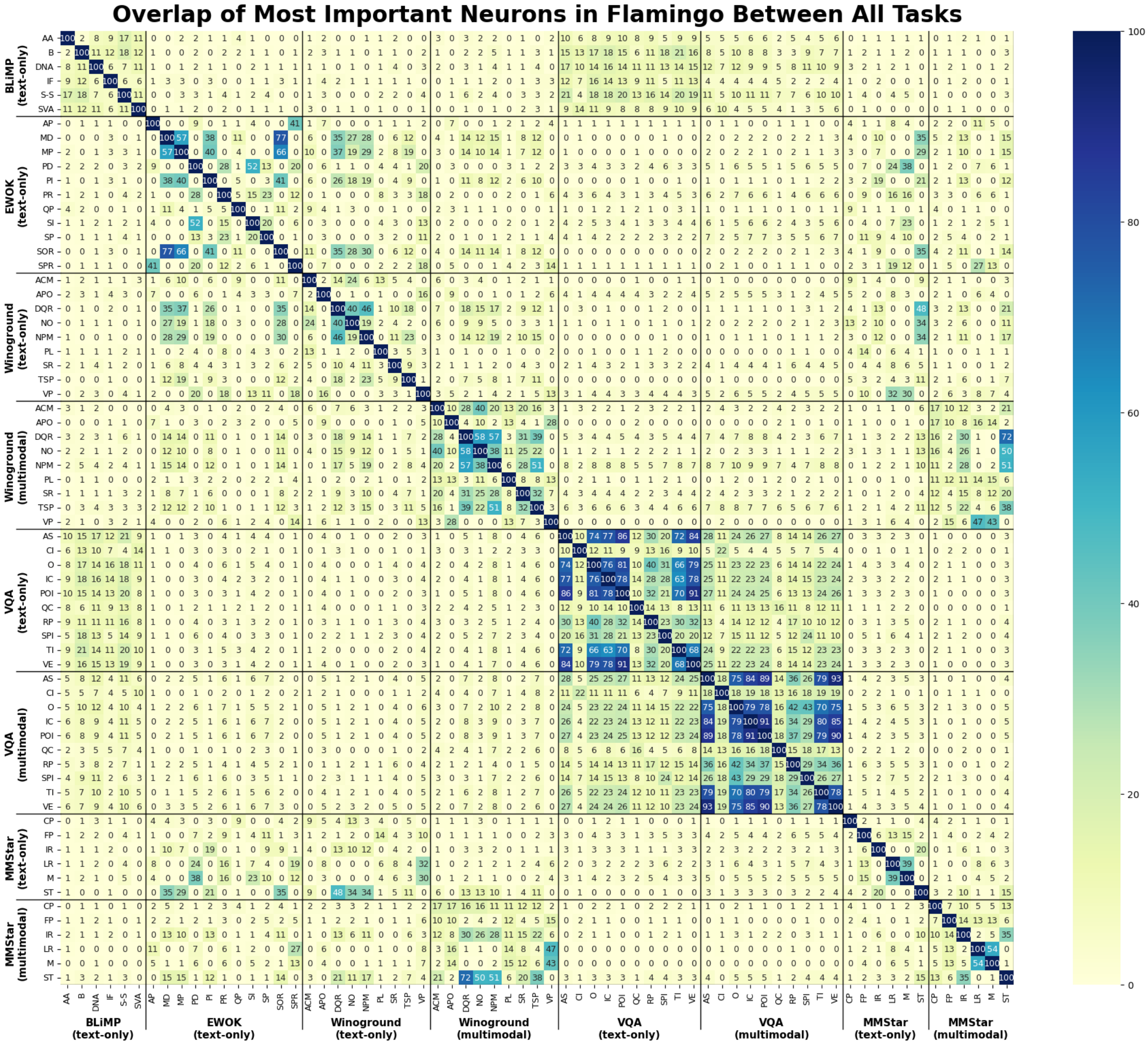


Figure 7: Overlap in Flamingo between the top 100 neurons by indirect effect per subtask for all benchmarks.

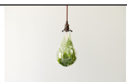
Benchmark	Prompt	Metric
BLiMP	The books about Galileo	`logit diff = final logit[token="are"] - final logit[token="is"]`
VQA	 Is this photo in color?	`logit diff = final logit[token="no"] - final logit[token="yes"]`
EWoK	Chao is making Yan's job easier. Chao is helping Yan. \n Chao is making Yan's job easier. Chao is hindering Yan.	`logit diff = logit sum["Chao is helping Yan"] - log sum["Chao is hindering Yan."]`
Winoground	 some plants surrounding a lightbulb \n a lightbulb surrounding some plants	`logit diff = logit sum[a lightbulb surrounding some plants] - logit sum[some plants surrounding a lightbulb]`
MMStar	 What is the main theme of the image? Transportation \n What is the main theme of the image? Outdoor recreation	`logit diff = logit sum[What is the main theme of the image? Transportation] - logit sum[What is the main theme of the image? Outdoor recreation]`

Table 7: Example prompts and their respective performance metric per benchmark used for attribution patching.

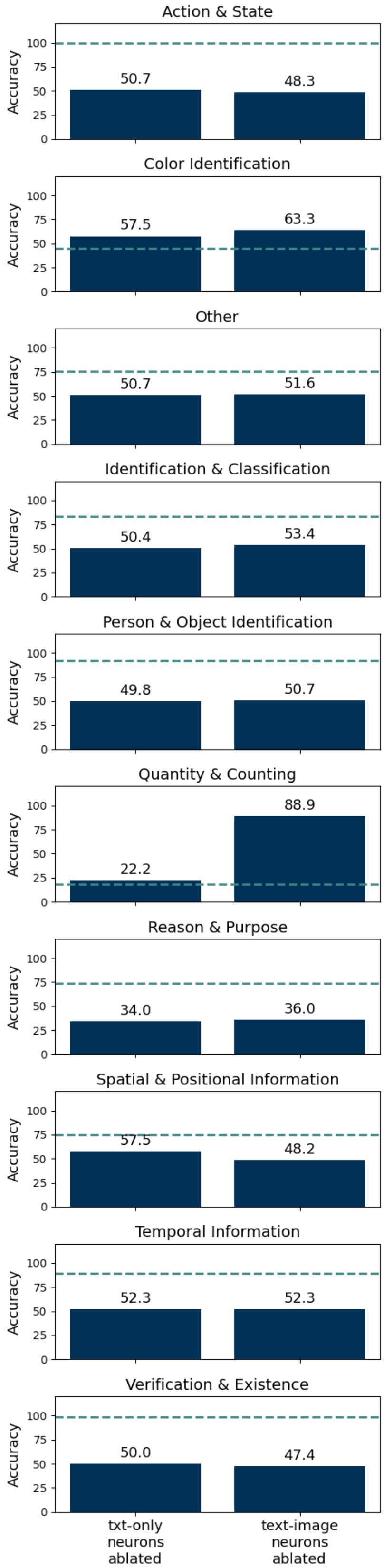


Figure 8: Clean and ablated GIT accuracy on VQA. Dashed line marks clean accuracy. The left and right bars show model performance without vision and with vision respectively.

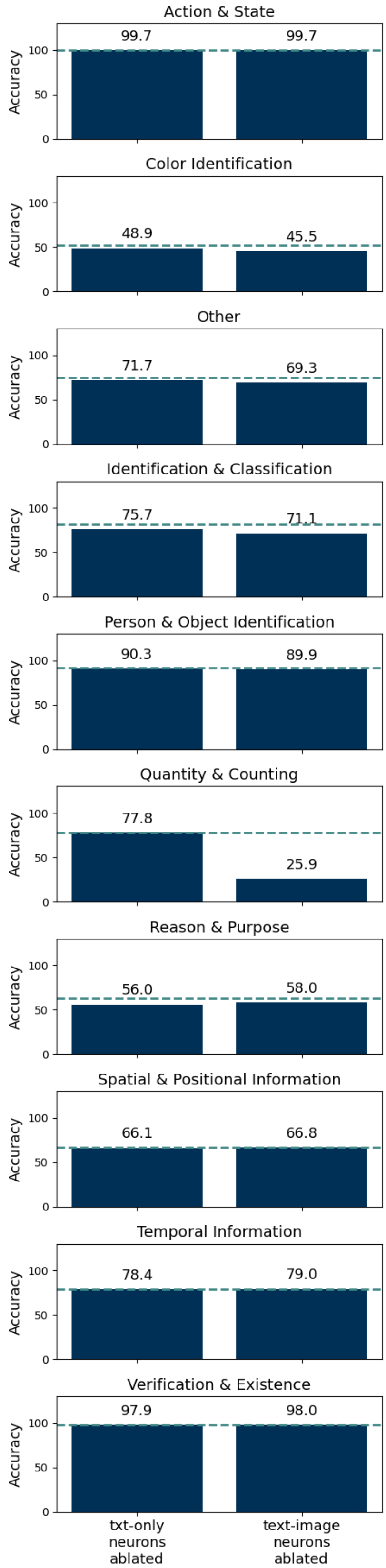


Figure 9: Clean and ablated Flamingo accuracy on VQA. Dashed line marks clean accuracy. The left and right bars show model performance without vision and with vision respectively.

ELC-ParserBERT: Low-Resource Language Modeling Utilizing a Parser Network With ELC-BERT

Rufus Behr
Research Computing, Northeastern University
`r.behr@northeastern.edu`

Abstract

This paper investigates the effect of including a parser network, which produces syntactic heights and distances to perform unsupervised parsing, in the Every Layer Counts BERT (ELC-BERT) architecture trained on 10M tokens for the 2024 BabyLM challenge. The parser network's inclusion in this setup shows little or no improvement over the ELC-BERT baseline for the BLiMP and GLUE evaluation, but, in particular domains of the EWoK evaluation framework, its inclusion shows promise for improvement and raises interesting questions about its effect on learning different concepts. [1]

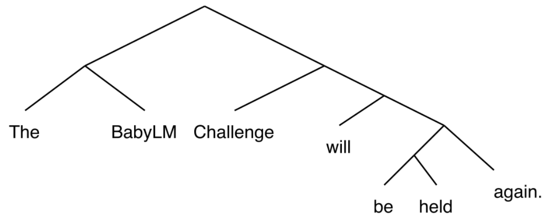


Figure 1: An example of an induced tree created from the model's unsupervised parser network

1 Introduction

Recent advancements in Transformer-based language models, in particular Large Language Models (LLMs), have largely been achieved by scaling the parameter count as well as the size of the dataset (Zhao et al., 2023). Whilst there is ongoing research in identifying efficient training and sampling methods for LLM pre-training, Villalobos et al. project that between the year 2026 and 2032 the datasets for training LLMs will be equivalent to all extant human text data.

In response to the staggering amount of data upon which LLMs are trained, the BabyLM challenge aims to incentive research in the development and pre-training of Language Models by setting realistic human-developmental limitations on the training data (Choshen et al., 2024). In particular, the challenge has three data-limited tracks: two texts only tracks that restrict the data corpora sizes to 10M and 100M (strict-small and strict, respectively), the latter of which is inspired by approximately the amount of data a 13 year old child will have seen, and a vision-language track, combining text and images.

[1]The code for the training and experimenting is available here: `https://github.com/SufurElite/ELC-ParserBERT`

The 2024 BabyLM Challenge is the second iteration of this challenge. The overall best system from the first challenge was the Every Layer Counts BERT model (ELC-BERT) (Georges Gabriel Charpentier and Samuel, 2023), which showed effective results by changing the residual connection between the transformer layers. Although, in the first BabyLM challenge, systems with architectural modifications produced the best results, a plurality of submitted systems used curriculum learning, of which only one found significant gain from this approach (Warstadt et al., 2023).

This paper introduces ELC-ParserBERT, a model submitted for the strict-small track, which incorporates the parser network proposed in (Shen et al., 2021) into ELC-BERT. The parser network is able to induce both dependency and constituency syntactic structures, an example of which can be seen in Figure 1, and the aim of its inclusion is to investigate whether this structural bias aids the baseline ELC-BERT model. This paper also investigates whether using a curriculum learning based approach with this model architecture yields any improvement.

2 Background Literature

Hu et al. propose a Transformer-based Syntactic Language Model (SLM), called Generative Pretrained Structured Transformers (GPST), that learns to induce syntactic parse trees in an unsu-

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 140–146
November 15-16, 2024 ©2024 Association for Computational Linguistics

pervised manner and is able to outperform GPT-2, including in the GLUE (Wang et al., 2018) evaluation dataset. In addition to the standard SLM with its Transformer backbone, the GPST has a composition component, a pruned inside-out encoder, namely, ReCAT (Hu et al., 2024b), which induces parse trees. The model is trained through a process akin to hard expectation-maximization: during the expectation stage, the model induces a parse tree from a compositional model, whose internal representation is used as input during the maximization stage that consists of updating all the GPST parameters.

The ReCAT component and its contextual inside-out layers made improvements upon unsupervised grammar induction when compared to the prior baselines (Hu et al., 2024b). One of the baselines it improved upon (both in terms of the F1 score for the syntactic trees and the memory complexity) was the StructFormer model (Shen et al., 2021).

The StructFormer also proposes an additional component, the parser network, that induces parse trees. Given its input of word sequences, the parser network generates syntactic heights and distances, which were proposed in (Luo et al., 2019) and (Shen et al., 2018), respectively. Given the syntactic heights and distances, the network then estimates the probability that a token is the head of another token. A directed weighted adjacency matrix is then created such that each weight is the probability a token depends on another. After these token-dependency relation probabilities are created in the parser network, they are used to constrain the self-attention (Shen et al., 2021). In addition to being evaluated on its unsupervised dependency and constituency parsing, the StructFormer was trained and evaluted as a masked language model.

As part of the first BabyLM Challenge, one system consisted of pre-training the original StructFormer architecture as well as variants of it (Momen et al., 2023). Their variants included using RoBERTa encoder in place of the standard transformer and at which layers to integrate the parser network – namely, placing the parser in the middle since there was supporting literature that shows syntactic information is better represented in the middle transformer layers. They concluded, however, that, although some of the evaluation tasks were improved upon by having a model that induces a syntactic bias into the architecture, there was not sufficient evidence that this inclusion improved the model architecture with respect to the

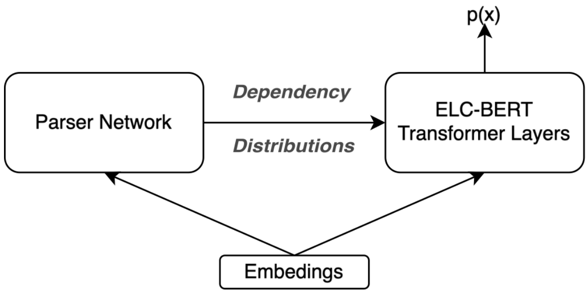


Figure 2: The model architecture

challenge set nor, within their experimentation, that the placement of the parser network in the middle of the transformer layers yielded improvement.

Another approach to inducing grammar induction is through compound probabilistic context-free grammars (compound-PCFGs) (Kim et al., 2019), wherein the model's context-free rule probabilities are alterable by a sentence-level latent variable. There was a submission to the first BabyLM challenge that made use of the compound-PCFG. The approach pre-trained a compound-PCFG on a subset of the strict-small training data and used the token embedding layer from the grammar as the input embedding layer to a different language model, which is subsequently trained on next work prediction on the training data (Chen and Portelance, 2023). They concluded, however, that there was no improvement over their baselines on account of the grammar induction, but that their choice of tokenizer, which was the WordPiece algorithm used to create both subword and whole word tokens, may have resulted in increased performance.

As mentioned above, the best submission to the first BabyLM challenge – and one of the baselines for this iteration – was the ELC-BERT (Georges Gabriel Charpentier and Samuel, 2023), which did not try to leverage syntactic structures but rather built upon the LTG-BERT model (Samuel et al., 2023) by introducing layer weighting.

3 Experimental Design

3.1 Model Architecture

Like the compound-PCFG system last year and the ELC-BERT model, a custom subword tokenizer was selected for the ELC-ParserBERT model, and it was trained on the provided strict-small data (Georges Gabriel Charpentier and Samuel, 2023; Chen and Portelance, 2023).

The model architecture in this paper uses the ELC-BERT architecture as its backbone (Georges

Gabriel Charpentier and Samuel, 2023) combined with StructFormer's parser network proposed (Shen et al., 2021) with the goal of increased performance from including both the layer weighting and the inductive bias from each, respectively. The architecture, therefore, follows that of the StructFormer but with weighted attention layers from ELC-BERT, as can be seen in Figure 2, where the Parser Network uses a combination of Convolutional layers, Linear layers, and the hyperbolic tangent function to produce the syntactic distances and heights that are used to compute the directed adjacency matrix with probabilities of a token depending on another.

3.2 Data

3.2.1 Training Data

The model uses the provided data from the organizers[2] for the strict-small track, which consists of the following: 8% from the dialogue portion of the British National Corpus (BNC) (Consortium, 2007); 29% from The CHILDES Project's database, a corpora of dialogue concerning child language (MacWhinney, 2000); 26% selected from the standardized Project Gutenberg, a corpus composed of over 50,000 books (Gerlach and Font-Clos, 2018); 20% from Open Subtitles, a corpus of subtitles extracted from movies and television (Lison and Tiedemann, 2016); 15% from nonfiction sections of Simple English Wikipedia[3], an encyclopedia written in plain English to be approachable for English language learners; and 1% from Switchboard, a corpus of dialogues made for dialogue act modeling(Stolcke et al., 2000).

The organizers ran initial preprocessing of the data to ensure that all the data was in plain text, but otherwise left preprocessing open to the contestants. The preprocessing of the training data for this model was largely inherited from the ELC-BERT (Georges Gabriel Charpentier and Samuel, 2023), where standardization is applied to the texts, the texts are compiled, split by line, and segmented into sentences using the Natural Language Toolkit's sentence tokenizer (Bird et al., 2009). After segmentation, the sentences are broken into sequence lengths, encoded by the model's subword tokenizer, and sorted according to their Flesch Reading Ease score (Kincaid et al., 1975) (to allow curriculum learning based upon this metric, if desired).

[2]The training data in its totality with references is available through OSF here: https://osf.io/ad7qg/

[3]https://dumps.wikimedia.org/simplewiki/

3.2.2 Evaluation Data

The model is evaluated on three evaluation benchmarks: BLiMP (as well as BLiMP supplemental) to evaluate the model's knowledge of grammatical phenomena (Warstadt et al., 2020); a selection of tasks that require finetuning from the General Language Understanding Evaluation (GLUE) and its more difficult successor SuperGLUE (Wang et al., 2020, 2018); and the Elements of World Knowledge (EWoK) framework, a benchmark that tests a model's world knowledge by examining the likelihood of context and target pairs across particular domains (Ivanova et al., 2024).

3.3 Experiments

There were two trained models with separate experimental purposes. The first trained model was the ELC-ParserBERT, trained on shuffled data that had a 15% probability of being masked, to be evaluated against the two provided baseline models for the strict-small track. The second model was the curriculum learning ELC-ParserBERT model (referred to hereafter as CL-ELC-ParserBERT), which also had a 15% probability of being masked and was presented in increasing Flesch Reading Ease (Kincaid et al., 1975), but it was compared against the submitted ELC-ParserBERT model in the EWoK evaluation framework. The hyperparameters for both models can be found in Appendix A. The scores were evaluated using the evaluation pipeline provided by the organizers[4].

When evaluating the LTG-BERT baseline model[5] locally, the scores achieved on the EWoK set were found to be different than the scores presented by the organizers. Henceforth, LTG-BERT-A refers to the scores presented by the organizers, and LTG-BERT-B refers to the scores evaluated locally.

Model	BLiMP	Suppl.	EWoK	GLUE	Macroaverage
BabyLlama	69.8	59.5	50.7	63.3	60.8
LTG-BERT-A	60.6	60.8	48.9	60.3	57.7
LTG-BERT-B	60.6	60.8	63.05	60.3	61.2
ELC-ParserBERT	59.6	57.7	63.1	44.5	56.2

Table 1: Model accuracies across different tasks

[4]The pipeline can be found here: https://github.com/babylm/evaluation-pipeline-2024/

[5]The model can be found here https://huggingface.co/babylm/ltgbert-10m-2024

Domains	ELC-ParserBERT	CL-ELC-ParserBERT	ELC-BERT-B
ewok_agent-properties_filtered	0.7376 ± 0.0094	0.7620 ± 0.0091	0.7552 ± 0.0091
ewok_material-dynamics_filtered	0.8104 ± 0.0141	0.8273 ± 0.0136	0.8740 ± 0.0120
ewok_material-properties_filtered	0.6000 ± 0.0377	0.4176 ± 0.0379	0.4647 ± 0.0384
ewok_physical-dynamics_filtered	0.3833 ± 0.0446	0.5083 ± 0.0458	0.3667 ± 0.0442
ewok_physical-interactions_filtered	0.5989 ± 0.0208	0.6025 ± 0.0208	0.6061 ± 0.0207
ewok_physical-relations_filtered	0.8166 ± 0.0135	0.8325 ± 0.0131	0.8166 ± 0.0135
ewok_quantitative-properties_filtered	0.4268 ± 0.0280	0.4013 ± 0.0277	0.5478 ± 0.0281
ewok_social-interactions_filtered	0.5646 ± 0.0290	0.5340 ± 0.0291	0.5374 ± 0.0291
ewok_social-properties_filtered	0.5610 ± 0.0274	0.4573 ± 0.0275	0.4451 ± 0.0275
ewok_social-relations_filtered	0.8068 ± 0.0100	0.7991 ± 0.0102	0.8036 ± 0.0101
ewok_spatial-relations_filtered	0.6347 ± 0.0218	0.6082 ± 0.0221	0.7184 ± 0.0203
ewok total score	0.6310 ± 0.0050	0.6136 ± 0.0050	0.6305 ± 0.0049

Table 2: A breakdown of the accuracies for ELC-ParserBERT, the Learning Curriculum ELC-ParserBERT, and the baseline ELC-BERT performs by each domain in the EWoK evaluation set.

4 Results

The results of the first experiment can be seen in Table 1. Although it performed poorly compared to the baselines in the (Super)GLUE evaluation and had slightly worse BLiMP supplemental scores, ELC-ParserBERT achieved comparable BLiMP scores to the LTG-BERT baselines and had significantly better scores on the EWoK evaluation framework than all other baselines, barring LTG-BERT-B.

Domain Name	p-val	Accuracy
ewok_material-properties_filtered_results	0.0170219	0.14
ewok_quantitative-properties_filtered_results	0.0017656	-0.12
ewok_social-properties_filtered_results	0.0017656	0.12
ewok_material-dynamics_filtered_results	5.97e-05	-0.06
ewok_spatial-relations_filtered_results	3.9e-06	-0.08

Table 3: EWoK domains where ELC-ParserBERT had significant difference in prediction from the LTG-BERT baseline, with the p-value and the change in accuracy relative to ELC-ParserBERT shown.

4.1 EWoK ELC-ParserBERT compared to LTG-BERT-B

ELC-ParserBERT's comparatively strong EWoK predictions prompted further analysis, namely, whether, although LTG-BERT-B and ELC-ParserBert had similar EWoK scores, there was any area where they had statistically significant different predictions. Upon preliminary inspection of the models' EWoK evaluation accuracies broken down by domain, as seen in Table 2, one can already see domains with disparate accuracies despite the close average score. To confirm that these are significant accuracy differences, however, a McNemar test can be constructed for each domain to determine the p-value for the difference in the models' classifications. In Table 3, the domains of the predictions that resulted in a p-value $< .05$ are listed.

These domains, however, can then be further broken down to see which categories within the domains had significant differences by running McNemar tests on the predictions for each group within a domain. In the "material properties" domain, ECL-ParserBERT predicted significantly better for the context type "direct" rather than "indirect," but it cannot be said that this is directly due to the inclusion of the structural bias in the model. It is more likely, however, that there are particular concepts that ELC-ParserBERT understands better or worse than LTG-BERT, although it may be possible that this is indirectly caused by the inclusion of the parser network in the model during pre-training. For instance, ELC-ParserBERT gets all 20 of the instances correct where the context is "cold" or "warm," whereas LTG-BERT only gets 4 of them right. The fourteen more that ELC-ParserBERT predicted correctly all had a "direct" context type, and there is a similar case for the "fragile" and "sturdy" contexts. Moreover, in the "quantitative properties" domain, ELC-ParserBERT actually performs worse in the "direct" context type questions, but this similarly follows from poor performance in particular concepts such as "a lot of" versus "a little."

The categories (concepts, context types, etc.) with significant differences in prediction between ELC-ParserBERT and LTG-BERT within the domains found to have significant differences, as seen

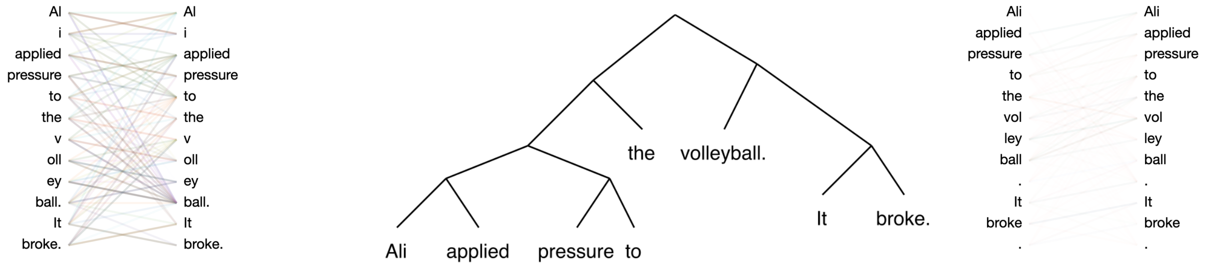


Figure 3: Given one of the contexts for EWoK, this figure shows an attention head of LTG-BERT on the left, the induced tree by ELC-ParserBERT in the middle, and an attention head of ELC-ParserBERT.

Domain Name	p-val	Accuracy
ewok_physical-dynamics_filtered_results	0.028784	-0.12
ewok_social-properties_filtered_results	0.0049673	0.10
ewok_agent-properties_filtered_results	0.0044958	-0.02
ewok_material-properties_filtered_results	0.0011381	0.18

Table 4: EWoK domains where ELC-ParserBERT had significant difference in prediction from CL-ELC-ParserBERT, with the p-value and the change in accuracy relative to ELC-ParserBERT shown.

in Table 3, are enumerated in full in Table 6, located in Appendix B.

4.2 Effectiveness of CL-ELC-ParserBERT

Similarly to LTG-BERT-B, CL-ELC-ParserBERT achieves comparable EWoK scores as seen in Table 2, and, when investigated further, there were four domains with significant difference in prediction between ELC-ParserBERT and CL-ELC-ParserBERT, as can be seen in Table 4. The most notable being the difference in the "material properties" domain, where CL-ELC-ParserBERT has an accuracy 18% smaller than ELC-ParserBERT. Interestingly, again, the concepts of "cold" and "warm" proved difficult for CL-ELC-ParserBERT in the same manner as it did for LTG-BERT-B. CL-ELC-ParserBERT also struggled with the concepts of "heavy" and "light," but it significantly outperformed ELC-ParserBERT when it came to concepts of "sink" and "float," as well as "fall" and "rise."

Although the final scores were close, when breaking down the scores into domains, it's interesting to see how the effects of curriculum learning rather than shuffling, in the context of this training data, result in significantly different predictions for certain domains.

4.3 Attention Comparison

To further examine how the inclusion of the parser network alters the model directly, one can see how the attention differs for a given input, as in Figure 3, by using BertViz (Vig, 2019). The weight of the lines connecting the tokens is based upon the attention between the words. Hence, the dampened weighting of the lines for ELC-ParserBERT shows how the attention is being constrained by the dependency relations produced by the parser network.

5 Conclusions and Future Work

In the context of the BabyLM Challenge 2024, this paper experimented with the ELC-ParserBERT architecture, which is formed by adding the parser network from the StructFormer (Shen et al., 2021) to the ELC-BERT architecture (Georges Gabriel Charpentier and Samuel, 2023). There was no significant improvement found in the BLiMP, BLiMP supplemental, and (Super)GLUE evaluation tasks through the inclusion of the parser network with the training as described. In the EWoK evaluation framework, however, the ELC-ParserBERT architecture showed comparable results to the LTG-BERT-B model and improvement over the other baselines.

This paper also examined the effectiveness of using the Flesch Reading Ease (Kincaid et al., 1975) metric to determine an ordering of the training data for curriculum learning for training the ELC-ParserBERT architecture. The use of this particular learning curriculum on this training data with this architecture did not show any significant improvement generally, but the inclusion or exclusion of this learning curriculum did significantly alter the quality of predictions for certain concepts. Investigating the cause of these particular concept affinities might be the focus for future work.

Future work may also seek to improve upon the ELC-ParserBERT model by ensuring sentences are producing parse trees separately for each sentence in a context window, as is done in GPST (Hu et al., 2024a). Additionally, the ELC-ParserBERT's largest shortcoming was in the (Super)GLUE evaluation tasks, which employed the default hyperparameters for finetuning set by the organizers, so searching for more optimal hyperparameters may yield overall model improvement.

Acknowledgments

This work was completed in part using the Discovery cluster, supported by Northeastern University's Research Computing team.

References

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media, Inc.

Xuanda Chen and Eva Portelance. 2023. Grammar induction pretraining for language modeling in low resource contexts. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 69–73, Singapore. Association for Computational Linguistics.

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. *Preprint*, arXiv:2404.06214.

BNC Consortium. 2007. British national corpus, XML edition. Oxford Text Archive.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every Layer Counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.

Martin Gerlach and Francesc Font-Clos. 2018. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *CoRR*, abs/1812.08092.

Xiang Hu, Pengyu Ji, Qingyang Zhu, Wei Wu, and Kewei Tu. 2024a. Generative Pretrained Structured Transformers: Unsupervised Syntactic Language Models at Scale. *Preprint*, arXiv:2403.08293.

Xiang Hu, Qingyang Zhu, Kewei Tu, and Wei Wu. 2024b. Augmenting transformers with recursively composed multi-grained representations. In *The Twelfth International Conference on Learning Representations*.

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 Akyürek, 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. *Preprint*, arXiv:2405.09605.

Yoon Kim, Chris Dyer, and Alexander Rush. 2019. Compound Probabilistic Context-Free Grammars for Grammar Induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2369–2385, Florence, Italy. Association for Computational Linguistics.

Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Hongyin Luo, Lan Jiang, Yonatan Belinkov, and James Glass. 2019. Improving Neural Language Models by Segmenting, Attending, and Predicting the Future. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1483–1493, Florence, Italy. Association for Computational Linguistics.

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Omar Momen, David Arps, and Laura Kallmeyer. 2023. Increasing the performance of cognitively inspired data-efficient language models via implicit structure building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 327–338, Singapore. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: Bert meets british national corpus. *Preprint*, arXiv:2303.09859.

Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. 2018. Straight to the Tree: Constituency Parsing with Neural Syntactic Distance. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1171–1180, Melbourne, Australia. Association for Computational Linguistics.

Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler, and Aaron Courville. 2021. StructFormer: Joint Unsupervised Induction of Dependency and Constituency Structure from Masked Language Modeling. *Preprint*, arXiv:2012.00857.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42, Florence, Italy. Association for Computational Linguistics.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Will we run out of data? limits of llm scaling based on human-generated data. *Preprint*, arXiv:2211.04325.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. *Preprint*, arXiv:1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *Preprint*, arXiv:2303.18223.

A Hyper Parameters

Hyperparameter	Value
Initial learning rate	5e-3
Batch size	256
Steps	13495
Attention probs dropout prob	0.1
Classifier dropout	0.2
Hidden dropout prob	0.1
Hidden size	384
Intermediate size	1024
Layer norm eps	1e-07
Max position embeddings	512
Num attention heads	6
Num hidden layers	12
Vocab size	16384
N parser layers	4
Parser conv size	9

Table 5: Hyperparameters used in the submitted model.

B EWoK Domain and Category Analysis

Domain Name	Category	p-val	Accuracy
material-dynamics	context type - direct	0.01174	-5.8%
material-dynamics	concept - wrinkle	0.00557	-11.7%
material-dynamics	context type - indirect	0.00253	- 6.8%
material-dynamics	concept - stir	0.00074	-8.0%
material-dynamics	target diff - concept swap	5.97e-05	- 6.4%
material-properties	concept - heavy/light	0.04123	37.5%
material-properties	target diff - concept swap	0.01702	13.5%
material-properties	context diff - antonym	0.01219	20.0%
material-properties	concept - cold/warm	0.00051	70.0%
material-properties	context type - direct	1.11e-05	38.5%
quantitative-properties	context type - direct	0.04228	-9.5%
quantitative-properties	concept - a lot of	0.02092	-26.0%
quantitative-properties	context type - indirect	0.01469	-18.5%
quantitative-properties	concept enough/not enough	0.00766	-25.0%
quantitative-properties	target diff - concept swap	0.00177	-12.1%
quantitative-properties	context diff - antonym	0.00103	-15.9%
social-properties	concept - friendly/hostile	0.03888	22.5%
social-properties	context type - indirect	0.01217	14.6%
social-properties	concept - tolerant/bigoted	0.00461	36.0%
social-properties	context diff - antonym	0.00369	12.6%
social-properties	target diff - concept swap	0.00177	11.6%
spatial-relations	context diff - antonym	0.01219	-5.3%
spatial-relations	target diff - concept swap	0.00842	-5.7%
spatial-relations	context diff - variable swap	4.40e-05	-16.4%
spatial-relations	context type - indirect	3.10e-05	-8.5%
spatial-relations	target diff - variable swap	3.04e-05	-15.0%
spatial-relations	concept - above/below	1.19e-07	-14.3%

Table 6: EWoK domains and categories of significant difference between ELC-ParserBERT and LTG-BERT with change in accuracy relative to ELC-ParserBERT.

Extending the BabyLM Initiative: Promoting Diversity in Datasets and Metrics through High-Quality Linguistic Corpora

Laurent Prévot
CNRS & MEAE
CEFC
Taipei, Taiwan
laurent.prevot@cnrs.fr

Sheng-Fu Wang
Academia Sinica
Institute of Linguistics
Taipei, Taiwan
sftwang@gate.sinica.edu.tw

Jou-An Chi
Graduate Institute of Linguistics
National Taiwan University
Taipei, Taiwan
r11142005@ntu.edu.tw

Shu-Kai Hsieh
Graduate Institute of Linguistics
National Taiwan University
Taipei, Taiwan
shukaihsieh@ntu.edu.tw

Abstract

BABYLM initiative paves the way for a range of experiments aimed at better understanding language models (LMs) and the differences and similarities between human and artificial language learning. However, the current framework is limited to the English language and a range of evaluation metrics, focused on syntax, semantics, and pragmatics. In this paper, we propose some steps towards extending the framework to other languages, like French, leveraging existing linguistic resources for these languages. Additionally, we advocate for greater exploration of genre variations within subcorpora for training LMs, as well as for the adoption of additional evaluation metrics with different underlying principles. Our proposal consists of using high-quality spontaneous speech corpora as a source for extracting production-related variables, which the models are then fine-tuned to predict. We hypothesize that these production-related features offer insights into the language processing mechanisms underlying the data and that cognitively sensitive models should outperform others in predicting these features. Specifically, we propose focusing on the prediction of phenomena such as speech reductions, prosodic prominences, sequences co-occurring with listeners' backchannels, and disfluencies. To illustrate our approach, we present an example involving the prediction of speech reductions and prosodic prominences in spontaneous speech in two different languages (French and English), using models trained on 10 million tokens from different data source mixtures.

1 Introduction

The BABYLM initiative is built on three interrelated aspects: (i) data sets for training language models, (ii) evaluation metrics designed to capture cognitive and linguistic skills and their development, and (iii) models that are either more cognitively plausible and/or capable of learning efficiently from "small" datasets. This initiative represents a strategic and timely effort to better understand the differences between artificial and human language learners.

While the 2023 edition focus was primarily on models, the 2024 call expands the scope to include investigations into both datasets and evaluation metrics—a crucial step, as we will argue in this position paper. Specifically, we propose concrete directions for expanding language model training datasets and exploring new evaluation metrics to deepen the linguistic and cognitive relevance of the BABYLM evaluation framework. Regarding evaluation metrics, we advocate for a novel approach that leverages existing high-quality spontaneous speech corpora.

One observation about the BABYLM initiative to date is that the datasets used are in English. While this is a natural starting point, it represents a significant limitation. Expanding the scope to include more languages is not only about better representing linguistic communities or potential model users. Achieving comparable, contrastive results across different languages within the BABYLM framework could offer valuable insights into both the learning models and the underlying learning processes.

While the original BABYLM initiative argues convincingly for a mix of data sources including transcripts of child-caregiver conversations, everyday conversations, subtitles, and simple texts, different mixtures can be explored. Due to data

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 147–158
November 15-16, 2024 ©2024 Association for Computational Linguistics

scarcity, it is still impossible to gather a 100M data set based on real spoken conversational data but the 10M is accessible for a few languages like English, French and Mandarin and a few others. Conversational speech is the genre within which humans acquire their basic language skills. It is a genre quite distant from the usual written or web content on which LMs are trained, increasing the risk of biases for LMs produced. Moreover, it has been argued that it is a genre of high relevance to language emergence (Levinson, 2020; Christiansen and Chater, 2022). How could a purely interactional dataset, including both child-directed and general conversation transcripts, be compared to more balanced mixtures? This opens the door for testing various hypotheses. For instance, does including more encyclopedic knowledge help with higher-level commonsense tasks, while a purely conversational training set provides a model with better communicative and conversational abilities?

In this context, current evaluation metrics, while a good starting point, appear biased in two ways: they tend to favor canonical written forms and prioritize syntactic, semantic, and commonsense pragmatics. However, language and communicative competence include many other dimensions. Although the initiative clearly emphasizes the importance of using speech transcripts, both child-directed and everyday conversations, as training data, to our knowledge, none of the evaluation metrics employed address explicitly the specificities of spontaneous speech.

To summarize, we argue that, in line with the directions proposed in this year's new call, training datasets, and evaluation metrics are just as crucial as models for understanding the computational learning of language structure. We propose evaluation metrics based on spontaneous speech data and demonstrate how we can build such metrics from different aspects of the speech signal and transcripts obtained from high-quality spontaneous speech corpora.

2 Related Work

Since the emergence of large language models, there has been strong interest from the computational linguistics community in understanding why they are so successful. Warstadt et al. (2020b) explore the conditions (e.g., the amount of training data) under which RoBERTa develops and leverages linguistic features, such as part of speech (POS) and morphology, as opposed to relying on simpler surface-level features like simple position-based or length-based features. More recently, several studies have probed LLMs to better characterize their performance across various domains, particularly with regard to their linguistic competence versus commonsense reasoning. These studies have also examined the relationship between model performance and the amount of training data required for different tasks. In particular, Zhang et al. (2021) used training sets of varying sizes, 1M, 10M, 100M, and 1B tokens, to show that syntactic and semantic competence becomes robust in the 10M-100M range, whereas larger datasets are needed to achieve strong results in pragmatic and commonsense reasoning tasks.

More broadly, there have been proposals for evaluating the performance of LLMs on diverse linguistic tasks. Warstadt et al. (2019b) leveraged a substantial body of generative syntax-semantics literature to develop benchmarks based on acceptability judgments, coming either the linguistic literature like the CoLA benchmark further extended by exploiting more sources and data augmentation methods in BLIMP (Warstadt et al., 2020a). In addition to these binary decision tasks, Zhang et al. (2021) combined three other types of evaluation metrics: *classifier probing* (following (Ettinger et al., 2016; Adi et al., 2017)), which includes tasks from POS tagging to coreference resolution; *information-theoretic* probing based on the minimum description length (MDL) principle; and *fine-tuning on higher-level tasks* such as those in the SuperGLUE benchmark.

Most of the benchmarks have been proposed for English. However, BLiMP Warstadt et al. (2019a) has inspired a series of language-specific benchmarks, such as CLiMP for Mandarin Chinese (Xiang et al., 2021), as well as benchmarks for other languages like Japanese (Someya and Oseki, 2023), Dutch (Suijkerbuijk et al.), and Russian (Taktasheva et al., 2024). These are important additions to the evaluation landscape. While these benchmarks represent important extensions to the general evaluation framework, they all rely on syntax-semantics structures derived from introspection and textbook data, as will be discussed in the next section. In parallel to these efforts, monolingual language models have been developed using large amounts of data (Chang et al., 2024), as well as experiments involving varied data quantities (Micheli et al., 2020).

In another line of research, several studies have tested the ability of large language models (LLMs) to perform tasks inspired by cognitive science, particularly in the domains of semantics and pragmatics (Ettinger, 2020; Binz and Schulz, 2023).

Our approach of using actual speech data to extract production-based metrics can be related to studies that use behavioral or neurophysiological data linked with linguistic datasets. Specifically, there has been significant work focusing on textual datasets combined with eye-tracking (Hollenstein et al., 2021) or neurophysiological (Bingel et al., 2016; Hollenstein et al., 2018) measures. Additionally, datasets from passive listening tasks, linked to fMRI, have been released for various languages (e.g., French, Mandarin, and English) (Li et al., 2022). These datasets have been used, for instance, to study the impact of training parameters on a language model's ability to predict neurophysiological data (Pasquiou et al., 2022). Focusing on spontaneous speech, (Rauchbauer et al., 2019; Hmamouche et al., 2024) examined the predictability of fMRI-derived signals from conversational variables, including lexical information.

In terms of specialized language models, (Cabiddu et al., 2025) developed LMs based on child-directed speech transcripts and evaluated them on word-sense disambiguation tasks. They concluded that word acquisition trajectories could be better captured by multimodal models that incorporate acoustic features, among other aspects. Regarding more specifically tokenizers, Beinborn and Pinter (2023) proposed an evaluation paradigm focusing on the cognitive plausibility of subword tokenization. They compared `BPE`, `WordPiece`, and `UnigramLM` and revealed a lower "cognitive correlation" for the latter. Lastly, in the most recent BabyLM edition, (Martinez et al., 2023) introduced an interesting learning curriculum that constrained vocabulary in the early stages to simulate more cognitively plausible learning curves. Although this approach did not yield consistent overall results, marginal gains were observed in selected tasks.

3 A proposal for a new source of metrics

All initiatives mentioned are grounded in text-based and/or handcrafted paradigms, potentially coupled with behavioral and /or physiological lab measures. In contrast, we propose using actual spontaneous conversational transcripts to build complementary benchmarks that test not only the syntax-semantics dimensions but also real-world language use. These metrics will remain fundamentally linguistic in nature rather than focusing on task-specific or end-to-end evaluation.

Language is acquired, especially in its early stages, within spontaneous, conversational environments. While conversational language shares grammatical structures with other genres, its unique characteristics suggest that simply listing syntactic "errors" or semantic incongruities does not fully capture linguistic competence. Furthermore, in a conversational context, what may be considered a production error from a formal grammatical perspective is often perfectly acceptable and successfully achieves its communicative purpose. Therefore, we aim to develop a complementary approach that provides a broader set of metrics for evaluating language models from both cognitive and communicative perspectives when combined with existing benchmarks.

Specifically, we propose using spontaneous speech corpora, as they offer insights into human language processing through various observable production phenomena. Our approach is a kind of *classifier probing* (Ettinger et al., 2016; Adi et al., 2017; Warstadt et al., 2019b), but rather than focusing on meta-linguistic tasks (e.g., predicting syntactic categories), we aim to predict phenomena that serve as partial indicators of language processing. We propose a preliminary set of potential metrics, which remains open for further development. These metrics include *speech reductions, listener's backchannel signals, prosodic prominences*, and *disfluencies*. The common point among these metrics is that they are all grounded in spontaneous speech production, and each has been the subject of extensive research.

3.1 Speech reductions

Speech reductions have been studied across a range of linguistic levels, from phonetics to semantics, especially when considering the issue of signal information density. In spontaneous speech, some chunks of speech are produced in a reduced manner, both in terms of duration and articulatory amplitude. The location of these reductions is not random. For example, studies have suggested that speakers tend to smooth the information density of their speech signal over time, with reductions serving as a mechanism to achieve this smoothing effect (Aylett and Turk, 2004).

The relationship between information density and speech reduction has led to research developments on this topic with various approaches. These approaches may differ in the probabilistic measures used to predict reductions, such as lexical frequency, contextual probability, and informativity (Aylett and Turk, 2004; Gahl, 2008; Cohen Priva, 2012; Seyfarth, 2014). They also differ in terms of the linguistic level at which reductions occur, whether at the phoneme-, syllable-, word-level, or in terms of overall speech rate. Many of these studies include and compare different types of probabilistic measurements (e.g., lexical frequency and contextual probability) within a single study (e.g., (Seyfarth, 2014; Cohen Priva and Jaeger, 2018)) and some of them also compare probabilistic measurements calculated at different linguistic levels (e.g., segment- and syllable-levels in Van Son et al. (1999), segment- and word/-level measurements in Van Son and Pols (2003), syllable- and word-level measurements in Wang (2022)). Inclusion and comparison of reductions or phonetic variability across various linguistic levels in the same study have also been done (e.g., individual segments and prefixes as a whole in Pluymaekers et al. (2005); morphemes and words in Tang and Bennett (2018)), albeit less frequently.

These studies show that phonetic reduction can be predicted to varying degrees on the basis of the statistical distribution of linguistic units, and the prediction has been repeatedly found with varying types of measurements at various levels of linguistic units. This motivates the development of a reduction-labeling task for evaluating language models.

3.2 Prosodic Prominences

Prosodic prominence refers to the emphasis placed on certain units, often demarcated at the level of words or syllables, within a spoken utterance. This emphasis can be measured through (and perceived based on) acoustic cues such as movements in fundamental frequencies, duration, intensity, and segmental properties such as the formant structure of vowels. Recent work by Wolf et al. (2023) has shown a significant degree of redundancy between the representations encoded from tokens alone and those derived from acoustic-prosodic information. Acoustic-prosodic features such as word-level energy, fundamental frequency, duration, pause, and composite measurements derived using a wavelet-based algorithm (Suni et al., 2017) were used to quantify this redundancy. Their findings suggest that prosodic information can be predicted, to some extent, from the word itself and its surrounding context.

Furthermore, Kakouros and O'Mahony (2023) suggests that language models (in their study, BERT) use syntax-semantics layers to predict prosodic aspects. While we do not argue that text alone can fully predict prosodic prominence (as also noted by Wolf et al. (2023)), we remark that part of prosodic prominence can indeed be predicted by a language model. In the case of spontaneous speech, this prosodic information reflects an additional layer of language processing. Therefore, language models that better capture human language processing should have an advantage over models trained exclusively on raw written text, particularly concerning prosodic prediction.

3.3 Listeners' Signals

Although not directly linked to the speaker's production, backchanneling (Yngve, 1970) offers another perspective on language processing. Backchannels do not occur randomly; they are frequent in casual conversations and closely related to turn-ending prediction (Skantze, 2021). There has been an ongoing debate about whether predicting the exact location of a turn-ending is a matter of lexical and syntactic completion prediction (De Ruiter et al., 2006) or based on prosodic cues from the main speaker (Bögels and Torreira, 2015). Finer-grained experiments by Riest et al. (2015) identified semantic completion as a crucial source of information for predicting turn-endings.

Our position is that if a listener can anticipate when it is appropriate to produce a backchannel, and even if part of this decision is based on prosodic cues from the main speaker, language models should be capable of predicting these moments to some extent.

3.4 Disfluencies

Directly predicting disfluencies (Shriberg, 1994), as discussed earlier, is challenging because disfluencies are explicit in the token stream. Removing them and labeling sequences where they originally appeared is cumbersome and potentially problematic. A more effective approach might be to adopt a well-established evaluation method: comparing "acceptable" versus "unacceptable" sequences. While disfluencies exhibit various subtleties, most follow a few simple patterns. We could compare actual utterances from high-quality linguistic corpora that do include detailed disfluency transcription with artificially generated utterances where disfluent patterns have been injected, similar to the syntactic acceptability approach used in Wagner et al. (2009) and Warstadt et al. (2019b).

4 Data

4.1 Pre-training data for creating LLMs

We have trained several language models. For the French experiment, we trained one model on 10M tokens from conversational datasets inspired by the original BabyLM data mix (ORFEO[1] (Benzitoun et al., 2016) and CHILDES-fr[2] (MacWhinney, 2014; Rose and MacWhinney, 2014)) and another on 10M tokens from Wikipedia. The training process used standard parameters (a BPE tokenizer with a 10K vocabulary size[3], a minimum token frequency of 2, and training for 3 epochs), with implementations from the HuggingFace packages.

Similarly, three English models were trained on three size-matched datasets containing 9M tokens from the following sources: a subset of the BabyLM 10M training data, "spoken" data that included BNC and Switchboard subsets from the BabyLM 100M training data, and a subset of Simple Wikipedia data from the BabyLM 100M training data. Subsets of the corresponding validation data from BabyLM were also used to create 0.9M-word validation sets for early stopping in LM training (maximum epochs = 100, early stopping patience = 3).

We included RoBERTa models in the fine-tuning experiments to serve as a topline for this task. The purpose of using RoBERTa models, which do not fit any of the BabyLM tracks, was to better contextualize our proposed metrics as a form of sanity check. The underlying idea is that if full-fledged LMs like RoBERTa fail to perform the task, it is likely that the task cannot be achieved given the provided data.

4.2 Benchmarks

For these experiments, we used two sources to build benchmarks: the Corpus of Interactional Data (CID) for French[4] (Blache et al., 2017) and the Buckeye Corpus for English[5] (Pitt et al., 2005). CID is an 8-hour corpus of 1-hour conversations between friends (16 speakers). It features fiercely spontaneous conversational speech. Buckeye is a corpus with 38.1 hours of spontaneous speech (40 speakers) recorded in an interview format.

The main reason for the choice of these corpora is the high quality of their speech transcript alignment, down to the syllable or even the segment level. These spontaneous datasets have also been used in various phonetic studies (Raymond et al., 2006; Meunier and Espesser, 2011).

5 Experiments

The experiments evaluated different pre-trained models for our set of tasks.[6] More precisely, we fine-tuned the pretrained models separately on a token classification task to predict which tokens were labeled (reduced / prominent / backchannelled) and which were not. A simple cross-validation was conducted across groups of speakers to maximize diversity across the folds.

5.1 Speech Reduction

There are several methods to determine whether a portion of speech is reduced. Following approaches in the literature, we first derived ratios of every word token's actual duration and its expected duration. For the French benchmark, we leveraged annotations of syllable boundaries in the corpus and developed a model that predicts syllable duration based on the segment it contains, similar to Wang (2022). A model is trained on one-half of the corpus and then applied to estimate the expected token duration in the remaining half of the corpus. For the English benchmark, we calculate words' expected duration from their component phonemes' mean duration in the corpus (Bell et al., 2009; Gahl et al., 2012; Seyfarth, 2014).

In both cases, we then converted the ratios into binary labels by applying a threshold of 0.7 (i.e., a reduction of at least 30%). This threshold resulted in labelling 33% of the tokens as reduced in the French benchmark and about 35% of the tokens in the English benchmark. These labels were then encoded in a BIO format.

[1] https://hdl.handle.net/11403/cefc-orfeo
[2] https://phon.talkbank.org/access/French/
[3] We tested vocabulary sizes of various sizes. Although the scores varied, they did not affect the performance hierarchy between the models.
[4] https://hdl.handle.net/11403/sldr000720
[5] https://buckeyecorpus.osu.edu/
[6] Notebooks for pretraining LMs and performing the experiment can be accessed at https://github.com/prevotlaurent/babyLM_TW_FR.

The main results for French and English are presented in Figures 1 and 2 respectively (see more detailed results in appendix). The results confirm that these models can predict speech reductions to some extent in a spontaneous speech corpus. Then, the "conversational" and "spoken" data models appear to have some advantages over the Wikipedia-based ones, even though the differences were not statistically significant.[7] Finally, the topline performance of RoBERTa is clear for the French results.

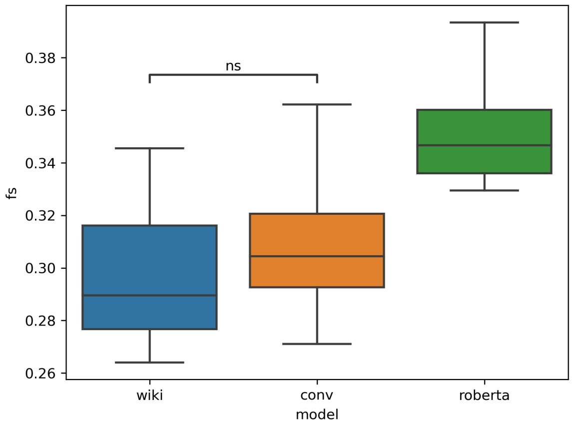


Figure 1: F-score comparing training data for predicting Speech reduction on CID corpus (RoBERTa as a top line). The significance between `wiki` and `conv` is not tested to be significant.

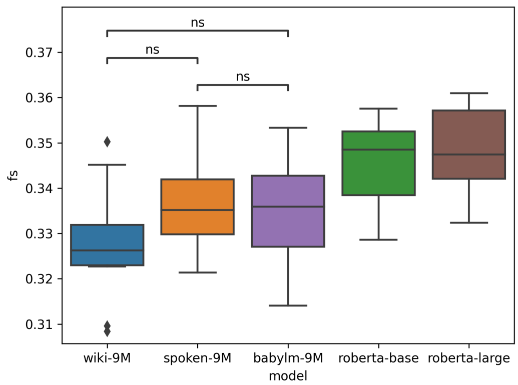


Figure 2: F-score comparing training data for predicting Speech reduction on the Buckeye corpus (RoBERTa models as top lines). The differences between models trained on 9M words were not significant.

5.2 Prosodic Prominences

To detect prosodically prominent tokens we used Suni et al.'s (2017) method based on wavelet that combines various acoustic features for determining prominence at the token level. One of the reasons for this tool choice is that it had been used already in the LMs literature (Wolf et al., 2023) to quantify the amount of redundancy between textual and prosodic levels. We used the default configuration of this tool and used a threshold score of 1.25 (See figure 9 in the appendix for details on the score values distribution). In the French data, this threshold amounted to 13.8% of the tokens labeled as prosodically prominent. In the English data, this threshold amounted to 14.7% of prosodically prominent tokens.

In both French and English experiments, the `conversational` and `spoken` models are significantly better than the `wiki` counterparts. RoBERTa models' topline performance is also clearer for both languages in this task.

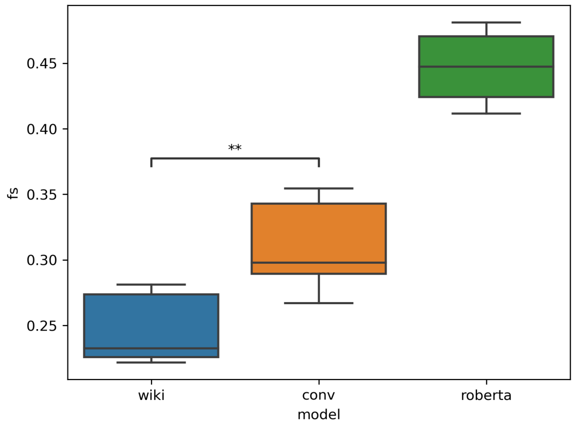


Figure 3: F-score comparing training data for predicting prosodically prominent tokens on CID corpus (RoBERTa as a sanity top line). **: p <= 0.01

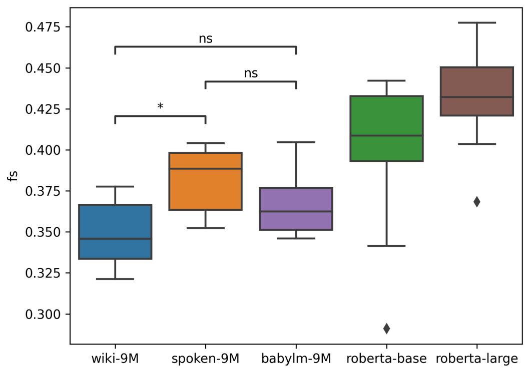


Figure 4: F-score comparing training data for predicting prosodically prominent tokens on the Buckeye corpus (RoBERTa models as top lines). *: p <= 0.05

[7] All statistical significances have been tested through a Mann-Whitney-Wilcoxon two-sided test.

5.3 Backchannels

We also designed, for the French benchmark only,[8] a task for predicting tokens around which a backchannel had been produced by the listener. To detect those, we used a simple list of tokens, eg. for French (*'mh', 'ouais', '@', 'ah', 'oui', 'bon', 'voilà', 'putain', 'accord', 'ben', 'oh', 'hum', 'eh', 'uh', 'OK'*). For each token of the target participant, we checked whether the other participant had produced one of these backchannel tokens in a time frame of $250ms$ before the beginning of the target token and $250ms$ after the end of the target token. This resulted in labeling 7.73% of tokens as being in the temporal vicinity of the listener's backchannels.

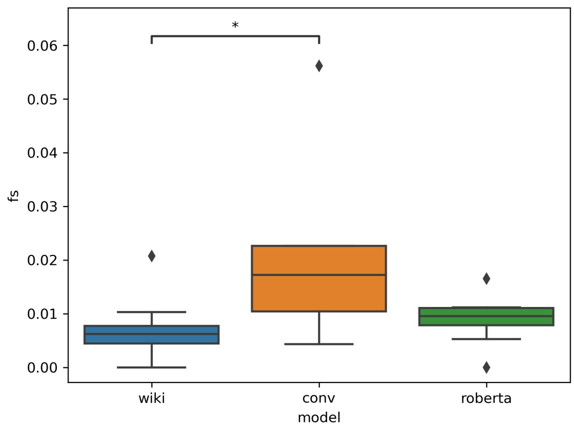


Figure 5: F-score comparing training data for predicting tokens overlapping a listener's backchannel in the CID corpus. *: p <= 0.05

As seen in figure 5, LLMs do not manage to solve this task with the data we gave them. While there is a statistically significant benefit for conversational pre-training (and in this case even over the bigger ROBERTA) the overall score does not go over 5% of f-score suggesting that none of these LLMs are getting close to modeling this phenomena. This is likely due to the nature of backchannelling: The literature points toward the contribution of lexico-syntactic cues to predict the end of turns, but the dominant cues remain prosodic ones, which these LLMs had no direct access to in their training data.

5.4 Testing models on BabyLM's zero-shot tasks

To examine whether models trained on spoken data can also be competitive in tasks that are not apparently tied to spoken language, we ran the English LMs[9] on the zero-shot classification tasks in BABYLM, i.e., BLIMP (Warstadt et al., 2020a) and EWoK (Ivanova et al., 2024), shown in Table 1. While the model trained on spoken data loses its advantage from our proposed reduction and prominence classification task and ranks the worst in the BLIMP supplement task, it is still competitive with other small models in filtered BLIMP and EWOK. Furthermore, the model trained on the BABYLM data, with a mixture of spoken and written materials, has the trend of outperforming the model trained on Simple Wikipedia both in our proposed tasks but also in BLIMP.

	BLIMP		EWoK
	supp.	filtered	filtered
ROBERTA-Large	71.9	73.9	65.5
ROBERTA-Base	70.3	74.3	62.9
BABYLM-9M	59.1	59.9	68.0
Wiki-9M	57.3	58.9	67.8
Spoken-9M	55.9	59.2	68.7

Table 1: English Models' performances in BLIMP & EWok

6 Potential shortcomings and Limitations

Information-centric nature. Our metrics are related to information-theoretic notions such as information density, entropy, and predictability. There is a substantial body of literature that demonstrates that these concepts can at least partially explain the phenomena discussed in the previous sections. This reminds us that information-theoretic measures, such as perplexity (a common LLM evaluation metric), are inherently connected to the variables we aim to predict. One potential limitation is that the models may only capture the information-theoretic contribution to our tasks. However, the prediction of these phenomena cannot be reduced to information-theoretic explanations alone. Each metric introduces its own set of subtleties related to language processing, and our goal is to evaluate LLMs in terms of their ability to grasp these subtleties.

Text-only. The phenomena we propose for probing the models are inherently related to speech processing, which goes beyond what

[8]This metric requires a truly conversational corpus with both parties accurately transcribed which is not the case of the English corpus used here.

[9]At the moment, we still lack similar benchmarks for French to do the same with our French LMs.

can be achieved with a text-only approach. Beyond the acoustic modality, the visual channel also plays a role, especially in contributing to backchannels. However, it is possible to limit multimodality to just text and speech by excluding face-to-face corpora from the benchmark. Our goal in proposing these metrics is not to achieve state-of-the-art performance in predicting these phenomena. Rather, we aim to treat them as *"traces"* of human language processing visible at the surface level, and to test which models are better at predicting these traces from text-only input.

Surface level shortcuts. A concern related to the previous point is the risk that models rely on surface-level elements as shortcuts to predict the variables we are targeting. While we do not have a definitive solution to this issue, since the nature of our metrics involves performance details observable in surface forms, we believe it is still worth pursuing this line of investigation. If the approach behaves consistently across our range of proposed metrics and languages, it may provide valuable information for language model evaluation. The next step will be to build controlled evaluation sets, similar to those developed in McCoy et al. (2019), that allow the exclusion of surface-level confounds in a principled way.

Triviality of the main result. From a machine-learning perspective, it might be seen as a trivial result that models trained on data similar to test sets perform better than models trained on other types of data. First of all, it is worth emphasizing that pretraining datasets and benchmarks in our experiments are completely independent as they do not come from the same raw corpora. Also, the pretraining datasets and corpora for building benchmarks have been curated by different teams and transcribed with different conventions. Nevertheless, we cannot deny that the conversational datasets are by all aspects (sentence length distribution, lexical frequencies, etc) more similar to benchmarks than Wikipedia datasets are.

As trivial as it seems, it may be one of our main points: to produce models more closely related to human cognition, one should use data sets made of spontaneous speech (and not generic textual / web content). The fact that RoBERTa outperforms all models does not change this fact since RoBERTa is trained on a dataset several orders of magnitude bigger.

7 Conclusion and Roadmap

In this position paper, we advocate for advancing the BabyLM initiative in several key areas. First, expanding beyond English is both necessary and feasible, given the initiative's design centered on "small-scale" data sets. Here we used French as an example, but we have also built the Mandarin equivalent datasets[10], emphasizing the importance of multilingual perspectives. Our proposal focuses on using training data composed entirely of spontaneous speech transcripts, which offers insights into language learning processes. It will be crucial to explore more nuanced variations in training data, such as balancing conversational speech, child-directed speech, and simple texts. Equally important is the development of complementary evaluation metrics. We propose using spontaneous speech data to benchmark models and assess linguistic phenomena, such as speech reductions, prosodic prominences, and backchannel responses, as key indicators of human language processing.

For the time being, we have English, Mandarin, and French training datasets with different data mixtures. The next steps involve systematizing the pilot experiments on speech reductions conducted here for the Mandarin dataset. Then, we will extract all the other proposed metrics for the benchmark datasets. Through this expanded set of experiments, we aim to demonstrate the value of the proposed approach and generalize it to other linguistic phenomena. In a broader perspective, we hope to show that benchmarks like BLIMP that require a significant amount of expert and naive human input to build, can be complemented with benchmarks derived from the numerous existing high-quality linguistic corpora, without additional human efforts.

Acknowledgments

We would like to thank Zoe Naud and Ri-Sheng for discussions in the context of this work.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of

[10]For Mandarin, we used Gutenberg, The NCCU Spoken Corpus of Spoken Taiwan Mandarin (Chui and Lai, 2008), Taiwan Corpus of Child Mandarin[11], Chinese Wikipedia and Open Subtitles.

sentence embeddings using auxiliary prediction tasks. In *Proceeddings of ICLR Conference Track*, Toulon, France.

Matthew Aylett and Alice Turk. 2004. The smooth signal redundancy hypothesis: A functional explanation for relationships between redundancy, prosodic prominence, and duration in spontaneous speech. *Language and speech*, 47(1):31–56.

Lisa Beinborn and Yuval Pinter. 2023. Analyzing cognitive plausibility of subword tokenization. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Alan Bell, Jason M Brenier, Michelle Gregory, Cynthia Girand, and Dan Jurafsky. 2009. Predictability effects on durations of content and function words in conversational English. *Journal of Memory and Language*, 60(1):92–111.

Christophe Benzitoun, Jeanne-Marie Debaisieux, and Henri-José Deulofeu. 2016. Le projet orféo: un corpus d'étude pour le français contemporain. *Corpus*, (15).

Joachim Bingel, Maria Barrett, and Anders Søgaard. 2016. Extracting token-level signals of syntactic processing from fmri-with an application to pos induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 747–755.

Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to understand gpt-3. *Proceedings of the National Academy of Sciences*, 120(6):e2218523120.

Philippe Blache, Roxane Bertrand, Gaëlle Ferré, Berthille Pallaud, Laurent Prévot, and Stéphane Rauzy. 2017. The corpus of interactional data: A large multimodal annotated resource. *Handbook of linguistic annotation*, pages 1323–1356.

Sara Bögels and Francisco Torreira. 2015. Listeners use intonational phrase boundaries to project turn ends in spoken interaction. *Journal of Phonetics*, 52:46–57.

Francesco Cabiddu, Mitja Nikolaus, and Abdellah Fourtassi. 2025. Comparing children and large language models in word sense disambiguation: Insights and challenges. *Language Development Research*, 5(1).

Tyler A Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K Bergen. 2024. Goldfish: Monolingual language models for 350 languages. *arXiv preprint arXiv:2408.10441*.

Morten H Christiansen and Nick Chater. 2022. *The language game: How improvisation created language and changed the world*. Random House.

Kawai Chui and Huei-ling Lai. 2008. The nccu corpus of spoken chinese: Mandarin, hakka, and southern min. *Taiwan Journal of Linguistics*, 6(2).

Uriel Cohen Priva. 2012. *Sign and signal: Deriving linguistic generalizations from information utility*. Ph.D. thesis, Stanford University.

Uriel Cohen Priva and T Florian Jaeger. 2018. The interdependence of frequency, predictability, and informativity in the segmental domain. *Linguistics Vanguard*, 4(s2).

Jan-Peter De Ruiter, Holger Mitterer, and Nick J Enfield. 2006. Projecting the end of a speaker's turn: A cognitive cornerstone of conversation. *Language*, 82(3):515–535.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. Probing for semantic evidence of composition by means of simple classification tasks. In *Proceedings of the 1st workshop on evaluating vector-space representations for nlp*, pages 134–139.

Susanne Gahl. 2008. Time and thyme are not homophones: The effect of lemma frequency on word durations in spontaneous speech. *Language*, 84(3):474–496.

Susanne Gahl, Yao Yao, and Keith Johnson. 2012. Why reduce? phonological neighborhood density and phonetic reduction in spontaneous speech. *Journal of memory and language*, 66(4):789–806.

Youssef Hmamouche, Magalie Ochs, Laurent Prévot, and Thierry Chaminade. 2024. Interpretable prediction of brain activity during conversations from multimodal behavioral signals. *Plos one*, 19(3):e0284342.

Nora Hollenstein, Emmanuele Chersoni, Cassandra L Jacobs, Yohei Oseki, Laurent Prévot, and Enrico Santus. 2021. Cmcl 2021 shared task on eye-tracking prediction. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 72–78.

Nora Hollenstein, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer. 2018. Zuco, a simultaneous eeg and eye-tracking resource for natural sentence reading. *Scientific data*, 5(1):1–13.

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

Sofoklis Kakouros and Johannah O'Mahony. 2023. What does bert learn about prosody? In *20th International Congress of Phonetic Sciences ICPhS*. International Phonetics Association.

Stephen C Levinson. 2020. On the human "interaction engine". In *Roots of human sociality*, pages 39–69. Routledge.

Jixing Li, Shohini Bhattasali, Shulin Zhang, Berta Franzluebbers, Wen-Ming Luh, R Nathan Spreng, Jonathan R Brennan, Yiming Yang, Christophe Pallier, and John Hale. 2022. Le petit prince multilingual naturalistic fmri corpus. *Scientific data*, 9(1):530.

Brian MacWhinney. 2014. *The CHILDES project: Tools for analyzing talk, Volume I: Transcription format and programs*. Psychology Press.

Richard Diehl Martinez, Hope McGovern, Zebulon Goriely, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. Climb–curriculum learning for infant-inspired model building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 112–127.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Christine Meunier and Robert Espesser. 2011. Vowel reduction in conversational speech in french: The role of lexical factors. *Journal of Phonetics*, 39(3):271–278.

Vincent Micheli, Martin d'Hoffschmidt, and François Fleuret. 2020. On the importance of pre-training data volume for compact language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7853–7858, Online. Association for Computational Linguistics.

Alexandre Pasquiou, Yair Lakretz, John T Hale, Bertrand Thirion, and Christophe Pallier. 2022. Neural language models are not born equal to fit brain data, but training helps. In *International Conference on Machine Learning*, pages 17499–17516. PMLR.

Mark A Pitt, Keith Johnson, Elizabeth Hume, Scott Kiesling, and William Raymond. 2005. The buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability. *Speech Communication*, 45(1):89–95.

Mark Pluymaekers, Mirjam Ernestus, and R Harald Baayen. 2005. Lexical frequency and acoustic reduction in spoken dutch. *The Journal of the Acoustical Society of America*, 118(4):2561–2569.

Birgit Rauchbauer, Bruno Nazarian, Morgane Bourhis, Magalie Ochs, Laurent Prévot, and Thierry Chaminade. 2019. Brain activity during reciprocal social interaction investigated using conversational robots as control condition. *Philosophical Transactions of the Royal Society B*, 374(1771):20180033.

William D Raymond, Robin Dautricourt, and Elizabeth Hume. 2006. Word-internal/t, d/deletion in spontaneous speech: Modeling the effects of extra-linguistic, lexical, and phonological factors. *Language variation and change*, 18(1):55–97.

Carina Riest, Annett B Jorschick, and Jan P de Ruiter. 2015. Anticipation in turn-taking: mechanisms and information sources. *Frontiers in psychology*, 6:89.

Yvan Rose and Brian MacWhinney. 2014. The phonbank project: Data and software-assisted methods for the study of phonology and phonological development.

Scott Seyfarth. 2014. Word informativity influences acoustic duration: Effects of contextual predictability on lexical representation. *Cognition*, 133(1):140–155.

Elizabeth Ellen Shriberg. 1994. *Preliminaries to a theory of speech disfluencies*. Ph.D. thesis, Citeseer.

Gabriel Skantze. 2021. Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, 67:101178.

Taiga Someya and Yohei Oseki. 2023. Jblimp: Japanese benchmark of linguistic minimal pairs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1581–1594.

Michelle Suijkerbuijk, Zoë Prins, Marianne de Heer Kloots, Willem Zuidema, and Stefan L Frank. Blimp-nl.

Antti Suni, Juraj Šimko, Daniel Aalto, and Martti Vainio. 2017. Hierarchical representation and estimation of prosody using continuous wavelet transform. *Computer Speech & Language*, 45:123–136.

Ekaterina Taktasheva, Maxim Bazhukov, Kirill Koncha, Alena Fenogenova, and Ekaterina Artemova. 2024. Rublimp: Russian benchmark of linguistic minimal pairs. *arXiv preprint arXiv:2406.19232*.

Kevin Tang and Ryan Bennett. 2018. Contextual predictability influences word and morpheme duration in a morphologically complex language (kaqchikel mayan). *The Journal of the Acoustical Society of America*, 144(2):997–1017.

RJJH Van Son and Louis CW Pols. 2003. How efficient is speech. In *Proceedings of the institute of phonetic sciences*, volume 25, pages 171–184.

RJJH Van Son, Louis CW Pols, et al. 1999. Effects of stress and lexical structure on speech efficiency. In *EUROSPEECH*.

Joachim Wagner, Jennifer Foster, and Josef van Genabith. 2009. Judging grammaticality: Experiments in sentence classification. *Calico Journal*, 26(3):474–490.

Sheng-Fu Wang. 2022. The interaction between predictability and pre-boundary lengthening on syllable duration in taiwan southern min. *Phonetica*, 79(4):315–352.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng Fu Wang, and Samuel R. Bowman. 2019a. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*.

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

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019b. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Lukas Wolf, Tiago Pimentel, Evelina Fedorenko, Ryan Cotterell, Alex Warstadt, Ethan Wilcox, and Tamar Regev. 2023. Quantifying the redundancy between prosody and text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9765–9784, Singapore. Association for Computational Linguistics.

Beilei Xiang, Changbing Yang, Yu Li, Alex Warstadt, and Katharina Kann. 2021. CLiMP: A benchmark for Chinese language model evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2784–2790, Online. Association for Computational Linguistics.

Victor H Yngve. 1970. On getting a word in edgewise. In *Papers from the sixth regional meeting Chicago Linguistic Society, April 16-18, 1970, Chicago Linguistic Society, Chicago*, pages 567–578.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1112–1125, Online. Association for Computational Linguistics.

A Variables to predict distributions and thresholds

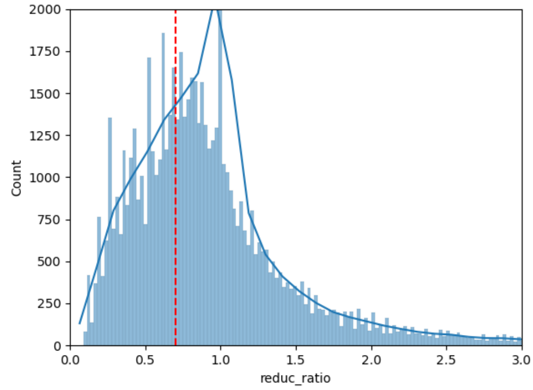


Figure 6: Distribution reduction ratios as calculated in the French Dataset and the threshold selected.

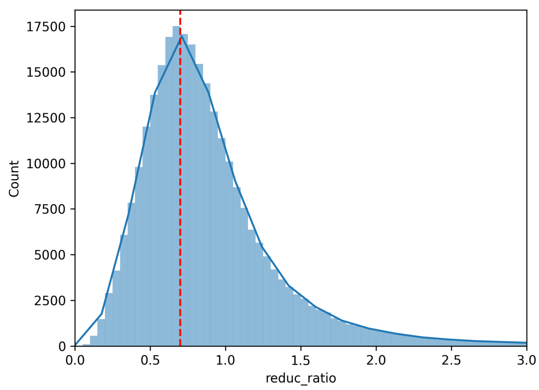


Figure 7: Distribution reduction ratios as calculated in the English Dataset and the threshold selected.

B Complete score tables

Language	Task	Model	F1	Precision	Recall
French	reduction	wiki-10M	.298 (.030)	.299 (.035)	.298 (.029)
		conv-10M	.310 (.029)	.300 (.033)	.321 (.030)
		XLM-Roberta-Base	.352 (.023)	.342 (.027)	.363 (.020)
	prominence	wiki-10M	.246 (.026)	.282 (.028)	.219 (.027)
		conv-10M	.311 (.033)	.356 (.039)	.277 (.033)
		XLM-Roberta-Base	.446 (.029	.503 (.040)	.403 (.033)
	backchannel	wiki-10M	.007 (.006)	.004 (.004)	.040 (.023)
		conv-10M	.020 (.016)	.014 (.014)	.057 (.025)
		XLM-Roberta-Base	.009 (.005)	.006 (.004)	.024 (.017)
English	reduction	Wiki-9M	.327 (.013)	.322 (.014)	.334 (.022)
		Spoken-9M	.336 (.012)	.333 (.011)	.340 (.019)
		BabyLM-9M	.335 (.012)	.331 (.013)	.340 (.020)
		Roberta-Base	.345 (.010)	.345 (.014)	.345 (.016)
		Roberta-Large	.349 (.009)	.343 (.011)	.355 (.015)
	prominence	Wiki-9M	.349 (.019)	.405 (.041)	.311 (.035)
		Spoken-9M	.382 (.020)	.453 (.046)	.333 (.028)
		BabyLM-9M	.366 (.018)	.437 (.045)	.318 (.029)
		Roberta-Base	.398 (.049)	.499 (.044)	.336 (.060)
		Roberta-Large	.431 (.030)	.488 (.057)	.392 (.046)

Table 2: Full results on the proposed speech-based benchmarks

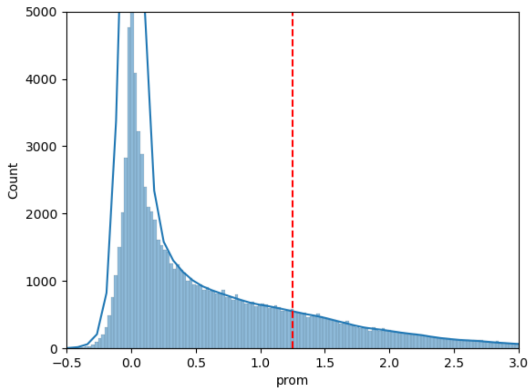


Figure 8: Distribution of prominence score as calculated in the French Dataset and the threshold selected.

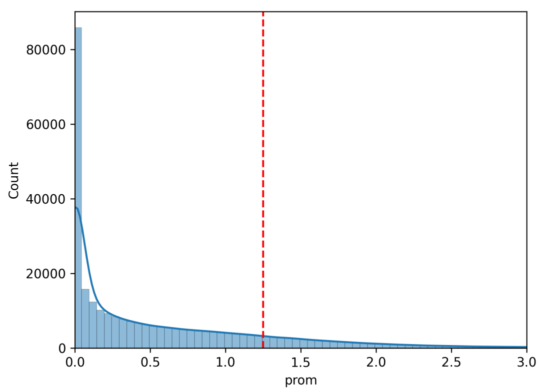


Figure 9: Distribution of prominence score as calculated in the English Dataset and the threshold selected.

Integrating Quasi-symbolic Conceptual Knowledge into Language Model Pre-training

Gábor Berend
Institute of Informatics,
University of Szeged
2 Árpád tér Szeged, Hungary
berendg@inf.u-szeged.hu

Abstract

In this paper, we investigate the integration of latent conceptual knowledge into the pre-training of masked language models. Our solution is based on the use of an auxiliary model, from which we extract training signals for training a student model. We determine the training signals from the hidden representations of the student model in an unsupervised way, using sparse coding. Models trained on latent concepts alone have an improved fine-tunability on downstream tasks, however, they perform worse on traditional language modeling, i.e., when the goal is to output missing tokens as opposed to latent semantic classes of words. In order to preserve the improved fine-tuning capability of the models, while making them better at the task of language modeling, we propose a final stage of pre-training, during which we perform traditional masked language modeling. The final stage of pre-training is based on a model that has already been pre-trained on the task of modeling latent semantic properties, with the weights of the backbone model being frozen. During the final training phase, we only train a lightweight linear classifier layer on top of the logits that the model determines for the latent semantic properties. With this modification, we can obtain the benefits of both the traditional training paradigms and the one which is based on the use of latent semantic properties. We release our source code at `github.com/SzegedAI/MLSM`.

1 Introduction

Language acquisition involves forming a rich battery of concepts and the ability to use and manipulate those concepts. Even though human cognition is rooted in concepts, this is not reflected in the typical pre-training of neural language models. In contrast, standard pre-training techniques ignore the concept-oriented nature of language when they expect a single ground truth token to be predicted during pre-training time.

Shani et al. (2023) argues for the need of integrating conceptual information into language models, while (Berend, 2023) recommended a knowledge distillation approach for doing so. The Masked Latent Semantic Modeling (MLSM) approach (Berend, 2023) relies on an auxiliary teacher model that steers the pre-training of the student model by performing sparse coding on its hidden representation and requiring the student model to recover those instead of the actual tokens. As the location of the non-zero coefficients in the sparse contextualized word representations obtained that way can be viewed as quasi-symbolic latent semantic concepts (Berend, 2020), the pre-training becomes driven by concepts as opposed to tokens.

While the favorable properties of MLSM pre-trained models have been demonstrated in obtaining models with improved fine-tuning capabilities, models pre-trained with it struggle on tasks that require language modeling ability, i.e., predicting actual token substitutes for missing/masked token positions from an input sequence. This is a consequence of the modeling in MLSM being shifted from the actual tokens to the latent concepts determined in an unsupervised way.

In this work, we extend such a modification to MLSM modeling, which ensures that the final model does not only have improved fine-tuning capability, but it is also capable of performing regular language modeling on the token level. We achieve this goal by integrating a lightweight post pre-training phase, during which we keep the weights of the model determined via MLSM fixed, and add a small a final linear module to the network (while freezing the rest of it), such that the token predictions are made on the logits that the originally pre-trained model would return towards the latent concepts. This modification ensures that the positive properties of MLSM and traditional masked language modeling (MLM) pre-training can be integrated into a single final model.

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 159–165
November 15-16, 2024 ©2024 Association for Computational Linguistics

2 Masked Latent Semantic Modeling

We first overview the MLSM pre-training technique, as it plays a central role in our modified model architecture. The way MLSM works is that it changes the domain of the output distribution of the model from its vocabulary of subword units (as in MLM) to the inventory of quasi-symbolic latent semantic properties that we determine in an unsupervised manner. In Figure 1, we provide a visual comparison between the MLM and MLSM pre-training techniques.

The way MLSM determines the latent semantic properties of some token is by relying on an already pre-trained auxiliary model $\mathcal{T}$. In a preparatory phase, a representative sample of hidden representations produced by $\mathcal{T}$ is collected from its layer l as $\{\boldsymbol{h}_1^{(l)}, \ldots, \boldsymbol{h}_N^{(l)}\}$. A dictionary learning problem (Mairal et al., 2009) is then solved of the form

$$\underset{\boldsymbol{D}^{(l)}, \boldsymbol{\alpha_j} \in \mathbb{R}^k_{\geq 0}}{\arg\min} \sum_{j=1}^{N} \frac{1}{2} \|\boldsymbol{h}_j^{(l)} - \boldsymbol{D}^{(l)} \boldsymbol{\alpha_j}\|_2^2 + \lambda \|\boldsymbol{\alpha_j}\|_1, \tag{1}$$

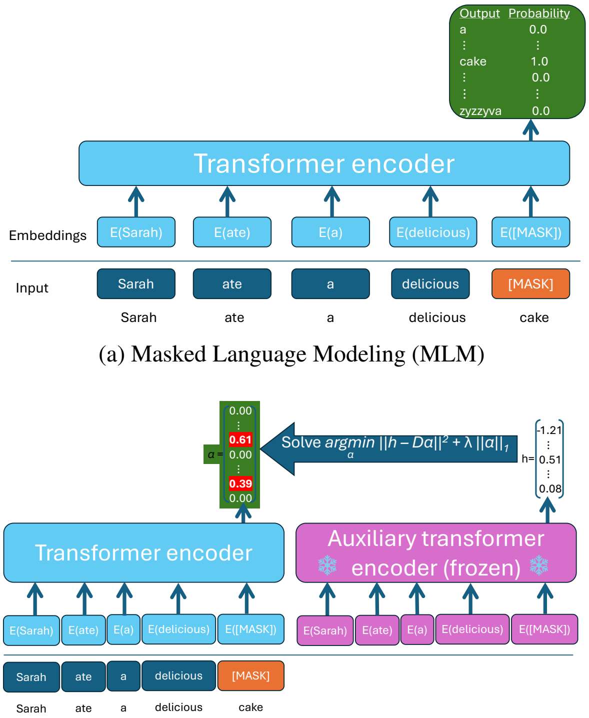
where $\boldsymbol{D}^{(l)} \in \mathbb{R}^{d \times k}$ is a dictionary matrix, with column vector norms bounded by 1, $\boldsymbol{\alpha_j} \in \mathbb{R}^k$ contains the sparse linear coefficients that indicate the extent to which the vectors from $\boldsymbol{D}^{(l)}$ are used in reconstructing the d-dimensional hidden representation from the l-th layer of $\mathcal{T}$, $\boldsymbol{h}_j^{(l)} \in \mathbb{R}^d$. λ serves as a regularization coefficient, controlling for the level of sparsity in $\boldsymbol{\alpha_j}$.

Solving (1) is performed in advance to the actual pre-training, with a negligible ($\ll 1\%$) computational overhead compared to the costs of pre-training. Once the dictionary matrix $\boldsymbol{D}^{(l)}$ is determined, it is used for determining the sparse contextualized representation for any $\boldsymbol{h}_i^{(l)}$, i.e., a hidden state from layer l of $\mathcal{T}$ as

$$\underset{\boldsymbol{\alpha_i} \in \mathbb{R}^k_{\geq 0}}{\arg\min} \frac{1}{2} \|\boldsymbol{h}_i^{(l)} - \boldsymbol{D}^{(l)} \boldsymbol{\alpha_i}\|_2^2 + \lambda \|\boldsymbol{\alpha_i}\|_1. \tag{2}$$

Objective (2) is computationally convenient, as it does not require optimizing towards $\boldsymbol{D}^{(l)}$. With $\boldsymbol{D}^{(l)}$ being fixed from (1), obtaining the sparse linear coefficients of $\boldsymbol{\alpha_i}$ constitutes an efficiently solvable LASSO optimization problem.

Due to the non-negativity constraint imposed towards $\boldsymbol{\alpha_i}$ in (2), the ℓ_1-normalized sparse linear coefficients can be conveniently treated as probability distributions over the k latent semantic concepts.



(a) Masked Language Modeling (MLM)

(b) Masked Latent Semantic Modeling (MLSM)

Figure 1: Comparison of the MLM and MLSM pre-training paradigms. The distributions in the green boxes represent the expected output for the masked token.

MLSM pre-training then considers these sparse normalized distributions of latent semantic concepts of the masked tokens as the desired target outputs and computes the Kullback–Leibler divergence as the loss function.

3 Pre-training

Our proposed pre-training consists of three sequential steps. In the first step, we used classical masked language modeling for pre-training. The models pre-trained at this stage serve as the auxiliary teacher model for the subsequently trained model (see Figure 1b). As MLSM does not output subtokens, it is expected to have limited capabilities in performing tasks that require outputting distributions over the vocabulary of the model.

We trained a separate Unigram tokenizer for the two corpora, with a 25,000 vocabulary size. During all three stages of pre-training, we used the AdamW optimizer with a peak learning rate of 0.0001 and an effective batch size of 1024 (that resulted from using gradient accumulation over 8 batches). When masking is involved, we employ the typically chosen 15% random masking rate selected dynamically from the batches.

3.1 Preliminary pre-training

The preliminary phase of pre-training was conducted using vanilla masked language modeling objective. At this stage, we trained a classical DeBERTa (He et al., 2021) model of the base size (i.e., 12 layers, 12 attention heads, 768 hidden dimensions). This model has roughly 100 million non-embedding parameters and approximately 20 million embedding parameters.

When pre-training the auxiliary models, we conducted 100,000 update steps, which roughly corresponds to 200 epochs on the 10 million token dataset, and 20 epochs on the 100 million training corpus. As it was the number of update steps that we kept constant, pre-training took roughly the same time on both corpora, i.e., approximately 2 days on a single NVIDIA A6000 GPU.

3.2 Pre-training involving latent concepts

Once the auxiliary model was created, we determined the dictionary matrix according to Eq. (1). We chose to extract $k = 1500$ quasi-symbolic latent properties based on the hidden representations originating from the last layer of the network ($l = 12$), using the regularization coefficient $\lambda = 0.05$. We selected the hidden representations from the auxiliary model for 1 million tokens from the respective corpora for determining the dictionary matrices.

The student models that we trained based on the dictionary matrices created in a preparatory phase were also DeBERTa base models. As the architecture of the student models are identical to the auxiliary model, it was possible to initialize the weights of the student models with those of the respective auxiliary model. Unless stated otherwise, we applied that kind of weight initialization of the student models.

For this phase, we went for an additional 20 epochs of pre-training. This corresponds to approximately 10,000 and 100,000 update steps for the 10 million and the 100 million pre-training corpora. This resulted in approximately 5 and 50 hours of additional GPU compute for the 10 million and the 100 million token corpora, respectively.

We also implemented such variants of MLSM that perform concept-driven pre-training without the need to employ special mask tokens. These variants are based on the observation that the range of input symbols in MLSM differs from that of the expected output symbols, i.e., the model receives subtoken units and outputs a distribution over k latent quasi-symbolic concepts, which renders masking during pre-training unnecessary.

The omission of masking has the benefit that we do not have to restrict ourselves to learning from only 15% of the input tokens (i.e., the ones that are masked otherwise), and it also makes the distribution of the sequences seen during pre-training more similar to the ones seen in either fine-tuning or inference time (due to the lack of a special mask token). Apart from not replacing 15% of the input symbols to a special mask token, this kind of pre-training is performed in the same way as MLSM, and we refer to this variant as Latent Semantic Modeling (LSM), reflecting the fact that no artificial masking token is involved during the pre-training.

We created two versions of LSM. One was such that it derived pre-training loss from all the input tokens, while the other version (the LSM15) is such that it omits the masking token during pre-training, but resembles typical pre-training which involves masking in that only a randomly selected 15% of the tokens is used for updating the model.

3.3 Language Modeling head training

The goal of this phase is to secure classical language modeling capabilities of the models that we pre-trained in the previous step using latent concepts. To achieve this goal, we take the resulting model from the second phase and add an extra linear module on top of it, the goal of which was to perform token predictions based on the logits that the model from the previous stage determined for the distinct latent semantic categories.

As we wanted to preserve the concept forming capabilities of the model and not alter its fine-tuning abilities, we froze all the weights of the backbone model, the only weights that were learned at this stage were the ones in the final, newly added linear layer, which transformed the k latent concepts to the vocabulary of the model. That is, we introduced an additional 1500×25000 parameters in order to improve the language modeling capability of our models, resulting in a final model of 158 million parameters (out of which 20 million were embedding parameters). As this phase of training only involved the calibration of a single linear layer, we opted for only 10 thousand update steps (corresponding to roughly 20 and 2 epochs on the 10M and the 100M token corpora, respectively).

corpus	Phase 1	Phase 2	Phase 3
strict-small	≈ 50 h	≈ 5 h	≈ 2 h
strict	≈ 50 h	≈ 50	≈ 2 h

(a) GPU hours (on an NVIDIA A6000)

corpus	Phase 1	Phase 2	Phase 3
stict-small	≈ 200	≈ 20	≈20
strict	≈20	≈ 20	≈2

(b) Epochs performed

Table 1: The amount of compute broken down at the individual phases.

This final phase took less than two hours of GPU calculation. In Table 1, we summarize the amount computation performed for arriving to a final model both in terms of GPU hours (Table 1a) and the number of epochs (Table 1b).

4 Experimental results

We evaluate our models using the official evaluation framework of the shared task that was provided by the organizers (Warstadt et al., 2023). The evaluation involved model fine-tuning on various GLUE tasks (Wang et al., 2019) as well as zero-shot evaluations towards the BLiMP (Warstadt et al., 2020) and EWoK (Ivanova et al., 2024) benchmarks.

4.1 Fine-tuning experiments

We did not investigated in hyperparameter optimization, simply adopted the default fine-tuning hyperparameters recommended by the organizers. The only hyperparameter we modified was the random seed of the fine-tuning, and we only modified it, so that we can report performances that are statistically more robust by averaging the fine-tuning performances obtained on the different tasks.

We repeated fine-tuning on all dataset 5 times (with random seeds ranging from 12 to 16) and report the mean performance on each task. The performance metrics we include are accuracy, except for the CoLA, MRPC and QQP tasks, where it is the Matthew Correlation Coefficient for the former, and the F1 score for the latter two. The averaged performance metrics are presented in Table 2. Those models that were additionally pre-trained with the objective of being able to predict the latent semantic properties of the tokens show better fine-tunability when trained on any of the pre-training corpora.

	MLM	MLSM	LSM15	LSM
BoolQ	0.665	0.669	0.668	**0.673**
CoLA	0.398	**0.417**	0.384	0.400
MNLI	0.757	**0.761**	0.758	0.760
MNLI-mm	0.764	**0.769**	0.768	0.765
MRPC	0.822	0.819	0.820	**0.823**
MultiRC	**0.646**	0.636	0.629	0.633
QNLI	0.828	0.831	0.833	**0.836**
QQP	0.861	0.862	**0.864**	0.863
RTE	0.535	0.545	**0.566**	0.564
SST2	0.893	**0.900**	0.896	0.892
WSC	0.415	**0.485**	0.462	0.392
Avg.	0.689	**0.699**	0.695	0.691

(a) models pre-trained on the 10M corpus

	MLM	MLSM	LSM15	LSM
BoolQ	0.686	**0.697**	0.689	0.693
CoLA	0.509	0.484	0.511	**0.541**
MNLI	0.779	**0.789**	0.782	0.783
MNLI-mm	0.783	**0.791**	0.788	0.790
MRPC	0.906	0.905	0.913	**0.919**
MultiRC	0.629	**0.643**	0.639	0.635
QNLI	0.846	**0.853**	0.849	0.852
QQP	0.868	0.868	0.869	**0.869**
RTE	0.616	0.607	**0.645**	0.632
SST2	0.903	**0.905**	0.899	0.898
WSC	0.400	**0.419**	0.412	0.396
Avg.	0.720	0.724	0.727	**0.728**

(b) models pre-trained on the 100M corpus

Table 2: Fine-tuning results of models pre-trained with different strategies. Results are the average of 5 independent experiments using random seeds ranging between 12 and 16.

Based on the results in Table 2, there seems to be little difference in the fine-tuning ability of the models that integrate latent semantic information during their pre-training (*LSM*), however, our next experiment reveals the true strength of the masking-free variants of MLSM. For this experiment, we started the latent semantics-driven pre-training of DeBERTa models with randomly initialized weights. In our previous experiments, the reason for being able to warm start our student model for the second phase of pre-training, i.e., to initialize it with the weights of the auxiliary model, was that the student and teacher models matched in both their architecture and size.

	MLM	MLSM	LSM15	LSM
BoolQ	0.665	0.640	0.677	**0.674**
CoLA	**0.398**	0.000	0.176	0.291
MNLI	**0.757**	0.347	0.750	0.755
MNLI-mm	**0.764**	0.342	0.756	0.762
MRPC	0.822	0.811	0.822	**0.826**
MultiRC	**0.646**	0.576	0.625	0.613
QNLI	**0.828**	0.509	0.818	0.815
QQP	**0.861**	0.000	0.854	0.855
RTE	0.535	0.460	**0.594**	0.573
SST2	0.893	0.518	0.882	**0.894**
WSC	0.415	**0.523**	0.392	0.439
Avg.	**0.689**	0.430	0.668	0.681

Table 3: Fine-tuning results of models pre-trained on the 10 million token corpus. Results are the average of 5 independent experiments using random seeds ranging between 12 and 16. This time the weights of the student models were randomly initialized and the pre-training of the student models involved only 10 million updates, while the auxiliary model was created in 100 million update steps.

It can, however, often be the case that the student model we train differs from the auxiliary in either of its size or architecture. In such cases, simply continuing the pre-training of the teacher model is not directly applicable. To this end, we conducted such experiments, where the student model – albeit remaining of the same size and architecture as the auxiliary model – was initialized with random weights, so that we can simulate a more general situation when continued pre-training is not an option to go for.

The results of this setting, when pre-training was conducted on the 10 million tokens strict-small dataset, is included in Table 3. We can see that the performance of MLSM degrades severely, whereas its masking-free counterparts do not degrade as much. In fact, the LSM pre-trained model manages to reach the performance of its teacher from a randomly initialized state in one tenth of the pre-training, as the second phase pre-training lasted only for 10 thousand update steps for the small-strict corpus, whereas we conducted 100 thousand training steps for obtaining the auxiliary model. We omit the results for the 100 million token corpus for brevity, but the general trends are the same in that case as well.

It is only the CoLA task, where the LSM model (initialized from scratch) lags behind the MLM pre-trained auxiliary model. This is not that surprising as the CoLA tasks is related to linguistic acceptability, for which task a model that was pre-trained to predict the correct word forms can offer better transfer compared to a model that was purely constructed to model latent semantic categories that arguably play a less important role when deciding linguistic acceptability.

4.2 Zero-shot experiments

We report next the results when evaluation is performed on the language modeling capabilities of the differently pre-trained models, i.e., the evaluation metrics on the BLiMP datasets (Warstadt et al., 2020) and the EWoK (Ivanova et al., 2024) benchmark. Table 4 contains the results of our auxiliary model, as well as our models prior going through the third phase of pre-training and after the final lightweight pre-training phase being completed.

It is not surprising that the models that were pre-trained with a focus on latent semantic categories are not performing well in language modeling prior to the final phase of pre-training. Table 4 reveals that once the final pre-training phase – which only involves training a single linear classification layer and is only conducted for 10K update steps – is finished, the models that were previously pre-trained with an emphasis on modeling latent semantic categories of tokens can perform just as well as the auxiliary model, which had a sole focus on being able to accurately predict masked word forms. As the weights of our backbone model were frozen during the last phase of pre-training, our models also preserved their ability to predict latent semantic categories to input tokens and the final token-level predictions are precisely made based on those categories determined by the models.

It is worth mentioning, that an alternative way to achieve that the trained models have a combined command of modeling latent semantic properties and concrete word forms would be the use of a multi-task objective, in which the MLM and MLSM objectives are combined together. Our preliminary experiments showed, however, that models pre-trained that way do not have better performance during fine-tuning. Moreover, this kind of multitask training objective would be incompatible with the masking-free variant of latent semantics based pre-training, as LSM does not replace any of the input tokens with a special mask token, something that is required by MLM pre-training.

	MLM	W/o third phase pre-training MLSM	LSM15	LSM	With third phase pre-training MLSM	LSM15	LSM
BLiMP	0.653	0.521	0.528	0.528	0.654	0.642	0.641
BLiMP supplement	0.603	0.511	0.484	0.494	0.590	0.580	0.591
EWoK	0.647	0.680	0.682	0.689	0.652	0.667	0.666
Average	0.634	0.571	0.565	0.570	0.632	0.630	0.633

(a) models pre-trained on the 10M corpus

	MLM	W/o third phase pre-training MLSM	LSM15	LSM	With third phase pre-training MLSM	LSM15	LSM
BLiMP	0.702	0.446	0.465	0.471	0.696	0.687	0.684
BLiMP supplement	0.623	0.495	0.529	0.533	0.654	0.613	0.608
EWoK	0.657	0.681	0.654	0.659	0.656	0.667	0.669
Average	0.661	0.541	0.549	0.554	0.669	0.656	0.654

(b) models pre-trained on the 100M corpus

Table 4: Zero-shot results of models pre-trained with different strategies.

4.3 Submitted results

In Table 5, we summarize the results that our submitted models achieved, along with the baseline scores provided by the shared task organizers, the BabyLlama (Timiryasov and Tastet, 2023) and the LTG-BERT (Samuel et al., 2023) models being the best performing decoder and encoder-based submissions in last years evaluation campaign.

	GLUE	BLiMP	BLiMP suppl.	EWoK
BabyLlama	0.633	0.698	0.595	0.507
LTG-BERT	0.603	0.606	0.608	0.489
MLSM	0.733	0.654	0.590	0.508
LSM15	0.721	0.642	0.580	0.508
LSM	0.708	0.641	0.591	0.507

(a) Using the 10M token strict-small pre-training corpus

	GLUE	BLiMP	BLiMP suppl.	EWoK
BabyLlama	0.690	0.731	0.606	0.521
LTG-BERT	0.684	0.692	0.665	0.519
MLSM	0.748	0.696	0.654	0.523
LSM15	0.747	0.687	0.613	0.527
LSM	0.741	0.684	0.608	0.522

(b) Using the 100M token strict pre-training corpus

Table 5: The baseline performances provided by the organizers and our final submitted scores, the results above the horizontal bars are the baselines provided by the organizers.

We can see a drop in the EWoK performances between Table 4 and Table 5. The reason behind this is that in Table 4, we reported evaluation metrics that we obtained using the official evaluation scripts during the development phase. The organizers, however, discovered that those scripts produced inflated scores on EWoK (which were caused by the way the evaluation framework handled ties in the probabilities produced by a model). The results in Table 5 are the ones that contain the EWoK scores after this issue has been fixed.

5 Conclusions

In this paper, we investigated the integration of latent concepts extracted from an auxiliary model into the sample efficient pre-training of neural language models. We gave multiple modifications to existing approaches, including a masking-free variant of the originally proposed approach and the inclusion of a final, lightweight pre-training phase into the pre-training procedure, which ensures that the final model is not only capable of modeling semantic properties of tokens, but it can also accurately predict the identity of masked word form based on the latent semantic properties that the backbone model determines. Finally, we make the models that we pre-trained openly accessible from `https://huggingface.co/SzegedAI` (the models named with prefix `babylm24`).

Acknowledgments

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of

Sciences. The research received additional support from the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory.

References

Gábor Berend. 2020. Sparsity makes sense: Word sense disambiguation using sparse contextualized word representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8498–8508, Online. Association for Computational Linguistics.

Gábor Berend. 2023. Masked latent semantic modeling: an efficient pre-training alternative to masked language modeling. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13949–13962, Toronto, Canada. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

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

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2009. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA. ACM.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Chen Shani, Jilles Vreeken, and Dafna Shahaf. 2023. Towards concept-aware large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13158–13170, Singapore. Association for Computational Linguistics.

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. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

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

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.

Are BabyLMs Second Language Learners?

Lukas Edman[1,2] **Lisa Bylinina**[3] **Faeze Ghorbanpour**[1,2] **Alexander Fraser**[2,4]

[1]Center for Information and Language Processing, LMU Munich
[2]Munich Center for Machine Learning
[3]Institute for Language Sciences, Utrecht University
[4]School of Computation, Information and Technology, TU Munich

lukas@cis.lmu.de, e.g.bylinina@uu.nl, faeze.ghorbanpour@lmu.de, alexander.fraser@tum.de

Abstract

This paper describes a linguistically-motivated approach to the 2024 edition of the BabyLM Challenge (Warstadt et al., 2023). Rather than pursuing a first language learning (L1) paradigm, we approach the challenge from a second language (L2) learning perspective. In L2 learning, there is a stronger focus on learning explicit linguistic information, such as grammatical notions, definitions of words or different ways of expressing a meaning. This makes L2 learning potentially more efficient and concise. We approximate this using data from Wiktionary, grammar examples either generated by an LLM or sourced from grammar books, and paraphrase data. We find that explicit information about word meaning (in our case, Wiktionary) does not boost model performance, while grammatical information can give a small improvement. The most impactful data ingredient is sentence paraphrases, with our two best models being trained on 1) a mix of paraphrase data and data from the BabyLM pretraining dataset, and 2) exclusively paraphrase data.

1 Introduction

Language models (LMs) need a lot of data in order to learn to approximate human linguistic behaviour (Warstadt and Bowman, 2022). The amounts of linguistic data typically used for training recent LMs is is significantly larger than what is available for most of languages of the world, and also much more than what children are typically exposed to during their first language acquisition. A 13 year old is typically exposed to less than 100 million words of linguistic input, which is orders of magnitude less than the amount used in LM pretraining. And still, LMs fail to be quite as good in language as human learners. Can we teach our models to be more data-efficient? If yes, how?

There are two potential strategies. One is to study how children acquire language in a natural setting, and use their acquisitional trajectories and patterns as inspiration for LM training. This intuition is one of the motivations for the BabyLM Challenge (hence the name; other low-resource pretraining contexts are, of course, also relevant): the challenge encourages LM pretraining optimization advancements inspired by human linguistic development (Warstadt et al., 2023).

Another direction is to embrace the obvious differences between LM pretraining and the ways human learners acquire their native language. The architectures of current LMs are dramatically different from human brain anatomy, and training objectives and strategies have only limited psycholinguistic developmental parallels. Finally – and most importantly for our contribution – input for first language acquisition by human learners and for LM pretraining is hardly comparable not only when it comes to dataset size. While the amount of strictly linguistic input that children get is small compared to typical LM training data, children get this input in communicative context that LMs lack at the pretraining stage, and it is typically paired with cross-modal data, which is not part of the `strict-small` track we choose for the BabyLM Challenge.

At a very high level, taking this second direction means that we look beyond human linguistic and cognitive development for optimization strategies – or at least, we do not need to expect that those will be the ones that necessarily work best.

We sharpen this point and contrast language learning in an acquisitionally realistic setting (first-language, or L1, acquisition) – and language learning in a more artificial setting – learning a second language, L2; a human activity that also leads to (different levels of) linguistic proficiency but contrasts dramatically with L1 acquisition by children. Almost everything is different: the set-up, the data, typical tasks the learner faces, and very often modality and their combinations.

We choose this particular direction mainly be-

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 166–173
November 15-16, 2024

cause in the current, second, edition, of the BabyLM Challenge participants are allowed to construct their own datasets within the track word budget. A lot of submissions last year, including ours, experimented with curriculum learning – different ways to order the same data (see our submission Edman and Bylinina (2023) as well as the BabyLM 2023 findings (Warstadt et al., 2023)). These attempts gave only limited results.

This year we instead focus on the effect of choosing different data on LM pretraining. In particular, roughly in line with how people learn foreign languages through explicit linguistic instruction, we divide training data into blocks roughly corresponding to types of linguistic information commonly found in English-as-a-foreign-language courses. We participate in the `strict-small` track allowing for only 10M words and experiment with four different types of linguistic information:

- **Lexical information** (information about word meaning and use), parallel to word learning in L2 acquisition. We use Wiktionary data as a source of this knowledge.
- **Grammatical information**, parallel to grammar learning for L2. We try two ways of constructing grammar data: a set of sentences marked with grammar phenomena, and texts of grammar books for L2 English learners.
- **Paraphrasing** has perhaps fewer obvious parallels in L2 learning practice, but is related to the explicit focus on sentential semantics ('different ways to say the same thing') and how different modifications in syntax and vocabulary can preserve and alter the meaning of a sentence, which is a common focus in L2 class discussions and exercises. For this data, we use one of the two SynSCE corpora from Zhang et al. (2021).
- A mix of **unconstrained textual data** that corresponds to various input during language acquisition of any kind, be it L1 or L2 acquisition. For this, we use portions of the BabyLM data provided by the challenge organizers.

We find that data on paraphrasing brings in the most significant improvements. Grammatical information is only marginally useful, even though it does come with some improvement, depending on the training set-up. Finally, lexical information does not seem useful for LM pretraining. One cannot be sure what to attribute these results to: the usefulness or lack thereof of particular types of data; the quality of the actual various datasets that we use; or the properties of evaluation used to judge whether a particular type of data is useful. One way or another, our answer to the question of whether BabyLMs are L2 learners is 'only when it comes to certain types of data'.

2 Data

2.1 BabyLM data

We make use of data provided by BabyLM organizers for our experiments. One of our two submitted models (Contr.) doesn't use BabyLM data at all, while the other one (Half/Half) uses a subset of BabyLM data. In the Half/Half model, we use the following parts of the BabyLM dataset:

Dataset	Words
Simple Wikipedia	145K
Gutenberg	254K
Switchboard	147K

Table 1: BabyLM data used for the Half/Half model.

We think BabyLM data roughly corresponds to unconstrained linguistic input in a language learner's experience (reading materials and practice conversations with language teachers and peers).

The rest of the data in the Half/Half model comes from the dataset we discuss next.

2.2 Contrastive dataset

An important part of the language acquisition experience is finding out how changes in phrasing and syntactic structure can alter or preserve meaning. This is seen in typical L2 learning tasks such as paraphrasing, which highlight the semantics of the sentence and the ways syntactic manipulation can affect its meaning.

As data approximating this type of information, we use a dataset by Zhang et al. (2021). They release two datasets as part of SynCSE, a contrastive learning framework for training sentence embeddings. The data in both datasets (SynCSE-partial and SynCSE-scratch) is synthetic: synthesized by LLMs. The two different datasets are results of different prompting set-ups (for the dataset construction and prompting details, we refer the reader to the original paper). We use one of these two datasets, SynSCE-partial[1].

The dataset is structured as follows: each datapoint comes as a triple consisting of 1) a sentence; 2) its paraphrase, and 3) a hard negative (a sentence that is similar to the original one lexically and/or structurally but has a different meaning). Here is an example of a triplet from the dataset:

> **sent0:** `One of our number will carry out your instructions minutely.`
>
> **sent1:** `One person from our group will execute your instructions with great attention to detail.`
>
> **hard_neg:** `Each member of our group will carry out your instructions differently.`

We use all three elements of the triplet in our experiments.

2.3 Grammar data

To mimic explicit grammar instruction in the typical L2 learning setting, we look for ways to expose the model to targeted grammatical information. We explore two strategies and corresponding datasets, which we call `Gram Gen` and `Gram Books`.

For **Gram Gen**[2], we first compile a list of grammatical notions that a sentence can contain. This list is inspired by the typical structure of reference and learners' grammars and the topics covered by those. We then pass these notions to GPT 4o-mini[3] to generate examples, using the prompt in Figure 1. To ensure that we generate a diverse set of sentences, we prompted the model to generate sentences about specific topics.[4]

After this, we additionally tag each sentence with the grammatical notions as a sentence can contain more than one. This again is done with GPT 4o-mini, using the prompt in Figure 2. Due to pricing restrictions, we generate 500 sentences per notion, and tag 100 of these sentences for 50 different notions. We include an example of a sentence tagged, where we verify the correctness of the given tags in Table 2.

In the table we can see that GPT 4o-mini appears only partially capable of recognizing grammatical notions. For the simpler, very well-known notions such as common nouns, verb person, tense, and number, GPT performs well. For less commonly-known phenomena, such as ellipsis, it seems to have no understanding. For ellipsis specifically, GPT often has false positives with sentences of this 2-clause structure, likely because that is a necessary component for an ellipsis to occur, but not what defines an ellipsis. GPT also appears to occasionally hallucinate, with "it" not appearing in the sentence despite it being tagged as an object pronoun. Overall, given the accuracy of GPT in tagging, it is not surprising that our model would struggle to grasp grammatical notions.

"The engineers proposed a new design for the bridge, while the architects focused on the aesthetic elements, emphasizing sustainability instead."

Notion	Tag	Correct?
common noun	engineers, design, bridge, architects, elements, sustainability	✓
collective noun	engineers, architects	✓ (partial)
singular noun	design	✓
plural noun	engineers, architects, elements	✓
nominative case	The engineers	✓
simple past tense	proposed, focused, emphasized	✓ (partial)
third person	engineers, architects	✓
plural verb	proposed, focused, emphasizing	✓ (partial)
indicative mood	proposed, focused, emphasizing	✓
non-gradable adjective	sustainable	✓ (partial)
positive adjective	sustainable	✗
aspectual adverb	emphasizing	✗
comparative adverb	instead	✗
object pronoun	it	✗
case preposition	for, on, instead	✓ (partial)
coordinating	while	✓
indefinite determiner	a new design	✓ (partial)
noun phrase	The engineers, a new design, the bridge, the architects, the aesthetic elements, sustainability	✓ (partial)
adjectival modification	aesthetic, sustainability	✓ (partial)
verb phrase	proposed, focused, emphasizing	✗
transitive verb phrase	proposed a new design, focused on the aesthetic elements, emphasizing sustainability	✓ (partial)
direct object	design, elements	✓
adjunct clause	Yes	✓
ellipsis gapping	Yes	✗
ellipsis pseudo-gapping	Yes	✗

Table 2: Tags produced for the sentence above. Only positive tags are shown for brevity. ✓ indicates the tag is completely correct, ✓ partially correct, ✗ incorrect.

We construct the second grammar dataset, **Gram Books**, as an alternative to grammatical instruction via examples. This dataset contains grammar books that overtly discuss the rules of English grammar and are intended mainly for second language learners of English. Here is the full list

[1] https://huggingface.co/datasets/hkust-nlp/SynCSE-partial-NLI

[2] We release this dataset on HF: `link placeholder`.

[3] We changed from GPT 3.5 to 4o-mini due to pricing changes.

[4] The possible topics are: accounting, anthropology, archaeology, architecture, art, artificial intelligence, astronomy, biology, botany, business, chemistry, computer science, cosmology, criminology, design, economics, education, environmental science, engineering, geography, geology, government, history, humanities, international relations, journalism, law, literature, linguistics, math, medicine, music, philosophy, physics, poetry, politics, psychology, religion, sports, and theater.

> You are an expert in grammar. Write 500 detailed sentences containing <notion> (as opposed to <alternate notion>). Make sure to write 500 detailed sentences that are all different from each other. Try to make the sentences sufficiently different, for example, don't start every sentence with "the", make both short and long sentences, and write about the topic of <topic>. Don't write anything else.

Figure 1: The prompt used to generate example sentences of a grammatical notion. The <alternate notion> is not always used, but corresponds to notions with clear alternatives, such as telic vs. atelic verbs.

> Consider the sentence: <sentence> Does the sentence contain the notion of <notion>? If so, write which word or words correspond to the notion. If not, write "N/A". Only write the word or words that correspond, or N/A otherwise.

Figure 2: The prompt used to tag sentences with their grammatical notion. The prompt for sentential notions only contained the initial question, along with: "Answer with yes or no. Only write 'yes' or 'no', nothing else."

of the grammar books we used: Newson (2006); Greenbaum and Nelson (2009); Roth and Aberson (2010); Thomson and Martinet (2015); Brutjan and Brutjan (2022); Wright (2024). We do not release this dataset due to copyright constraints.

We use both grammar datasets for two types of experiments: 1) regular MLM training (described in Section 3.2); 2) more elaborate training schemes involving a combination of an encoder and a decoder (discussed in Section 3.3).

2.4 Wiktionary

For lexical instruction, we make use of a segment of data from Wiktionary[5], the largest available collaborative source of lexical knowledge. We constrain ourselves to the English segment of Wiktionary, and extract the lemma together with parts of speech and the definitions of each of its senses and examples that illustrate the senses.

We parse the Wiktionary data into CSV, where each row contains a word, part of speech, a definition, and up to 13 examples, though many contained no examples.

> Give 3 examples of the word <word> as a(n) <part of speech>, where it means <definition>. List the 3 examples in a numbered list, they should be full sentences. Don't say anything else. The format should look like:
> 1. Example 1
> 2. Example 2
> 3. Example 3

Figure 3: The prompt used to generate example sentences of a word sense.

For words without an example, we attempted two things: we generated examples with GPT 3.5, and we fed the word in as is. The examples generated were of notably high quality, with GPT even able to generate sentences for rare word senses. The prompt we used is shown in 3.

As with other types of linguistic knowledge, with this data we are looking for a way to mimic typical L2 learning. Wiktionary comes pretty close to word learning in this setting, as it contains explicit information about different senses of the word, its morphological and syntactic profile, defines its lexical semantics and illustrates all of this information with sentences where the word is used in its different senses.

Again, as with grammar data, we use the resulting Wiktionary dataset[6] both in experiments with simple MLM pretraining and in experiments with more complicated training set-ups, which are described in more detail in Sections 3.2 and 3.3, respectively.

3 Method

3.1 Model Choice

We opted to use encoder-only models for our final submission. This is based on our observation from last year's competition, where encoder-only models generally outperformed decoder-only or encoder-decoder models. We chose the DeBERTa-base (He et al., 2021) architecture as it is considered state-of-the-art for encoder-only models. Unlike in last year's competition where we saw improvements

[5] http://www.wiktionary.org/

[6] The dataset we construct is available on HF: `link placeholder`.

from using DeBERTa-large, we saw no improvement this year in initial testing and thus only used the base model size.

3.2 Training and Evaluation

Our pretraining uses the standard MLM scheme (Liu et al., 2019), which we used last year to great effect. Table 3 shows the hyperparameters we used for our pretraining experiments. For fine-tuning, we use the default hyperparameters provided by the organizers.

Hyperparameter	Value
Vocabulary size	40000
Context size	64
Learning rate	2e-4
Decay	0.01
Warmup steps	4000
Optimizer	AdamW
Batch size	64, 256
Epochs	50

Table 3: Hyperparameters used.

The hyperparameters chosen are largely the same as what we used in last year's competition (Edman and Bylinina, 2023), with some minor changes to the learning rate (2e-4 vs. 1e-4) and warmup steps (4000 vs. 10000), as well as using both a batch size of 64 and 256. We found that, in some circumstances, a batch size of 64 would result in a more performant model, but this phenomenon was inconsistent. As such, we report the best performing batch size for each model. We note that "context size" refers to the number of tokens in a given example. This is constant, so each example may contain multiple sentences or fragments.

We evaluate our models with the tasks included in this year's shared task: BLiMP (Warstadt et al., 2020), BLiMP supplement, (Super-)GLUE (Wang et al., 2018, 2019), and EWoK (Ivanova et al., 2024).

3.3 Additional Training Schemes

In addition to using encoder-only MLM training, we experimented with other objectives to train using our Wiktionary and grammar data, but ultimately found no discernible difference in performance. For these experiments, we use an encoder-decoder model, where the decoder is later removed after training. The encoder part is simultaneously

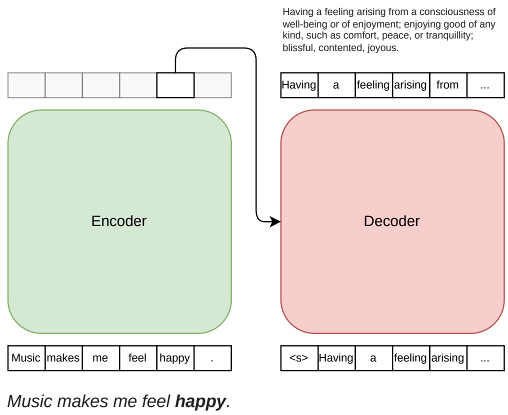


Figure 4: The model layout for training wiktionary.

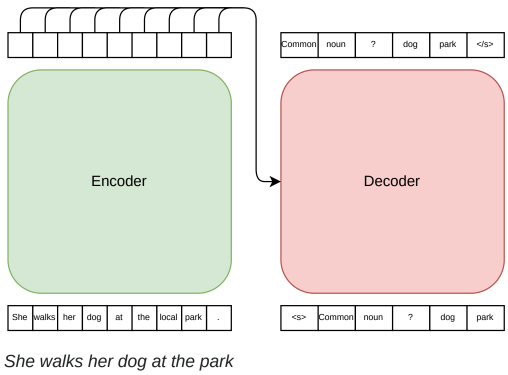


Figure 5: The model layout for training with grammar examples.

trained on MLM as well as the additional objectives, which we now describe.

Wiktionary Training For each Wiktionary entry, we feed the example as input to the encoder and mark the specific token that corresponded to the target word. For the marked position, we pass this to a separate decoder, which is tasked with generating the definition. This process can be seen in Figure 4.

Grammar Training For the `Gram Gen` data, we feed in a sentence to the encoder, passing its hidden states to the decoder, and prompt the model to answer whether it contains a particular notion, and if that notion corresponds to a particular word or words, which word(s) does it correspond to. The scheme for training is shown in Figure 5.

4 Results

We first discuss the results of our experiments with MLM-only models trained on grammar and lexical data, then we move on to discuss the results of the models with additional training schemes. Finally, we cover the results of our best-performing models that we submitted to the challenge.

4.1 Grammar Results

The results for our best models using grammar data are shown in Table 4. As we can see, adding grammar data appears to help with BLiMP to a limited extent, but hurts performance on all other metrics. The increase in BLiMP is expected, as the BLiMP evaluation necessitates that grammatical sentences are given a lower perplexity than ungrammatical sentences. A lot of the sentences in BLiMP are grammatical, but are very unnatural for a native speaker to read. As such, an excellent source for unnatural sounding yet grammatically correct sentences is a grammar book. This is likely why we see the most improvement from training on those.

The generated data, seeing as it is generated by GPT 3.5, is likely going to reflect the data that GPT itself was trained on. Although we do not know specifically the data that GPT is trained on, it is likely much more representative of "natural" data, rather than these unnaturally constructed sentences that are ubiquitous in BLiMP.

	Half / Half	+ Gram Gen	+ Gram Books
BLiMP	74.2	74.7	**75.4**
Supplement	**63.7**	63.3	61.1
GLUE	**77.1**	75.9	74.7
EWoK	**54.3**	53.0	50.3
Average	**67.3**	66.7	65.4

Table 4: Results of our grammar-informed models.

To further improve BLiMP scores, we expect that including more grammar books or perhaps explicitly prompting an LLM to produce unnatural sounding sentences may be the key. However, we also expect that such data would have a negative impact on GLUE and EWoK. This may simply be an immutable trade-off for low-resource pretrained models.

4.2 Wiktionary Results

We show the results of adding Wiktionary data in Table 5. Unfortunately, adding Wiktionary definitions and examples appears to only hurt performance. We speculate that it might have to do with the structure of Wiktionary entries and how the structure of lexical information is drastically different from other types of training and evaluation data.

	Half / Half	+ Wikt
BLiMP	**74.2**	72.9
Supplement	**63.7**	62.8
GLUE	**77.1**	75.7
EWoK	**54.3**	50.1
Average	**67.3**	65.4

Table 5: Results of adding Wiktionary data.

4.3 Additional Training Schemes Results

	MLM	MLM + Gram	MLM + Wikt
BLiMP	74.2	71.5	**75.7**
Supplement	**63.7**	61.0	59.3
GLUE	**77.1**	75.9	73.4
EWoK	**54.3**	51.1	50.8
Average	**67.3**	64.9	64.8

Table 6: Our models with additional objectives, compared to the MLM-only baseline (i.e. our half/half model).

We show the results of our models with added objectives for Wiktionary definition learning and grammatical notion identification in Table 6. Concerning the grammar objective, we see slightly worse performance overall. Notably, despite BLiMP being an evaluation aimed at gauging understanding of grammaticality, we still see a decrease in the performance.

Ironically, our Wiktionary-based objective increases BLiMP scores. It is unclear why our method for improving semantic understanding increased performance on the grammar benchmark, but there is of course information that can be extracted from word definitions that is useful for parsing grammaticality, such as part of speech information, and even quite literal information about the usage of words (e.g. the definition of "the" starts with "used before a noun phrase...").

Though it does not explain the improvement on BLiMP from our model trained with the Wiktionary objective, we believe that adding an additional objective is the main source of the loss in performance for our additional models. BLiMP (as well as EWoK) is designed such that a model's zero-shot default behavior is to provide a perplexity for a sentence. This is achieved trivially with a model trained on MLM or CLM, but adding another objective means that the hidden states are forced to

	BabyLlama		LTG-BERT		BabyLM	Half / Half	Contr.
	10M	100M	10M	100M	10M	10M	10M
BLiMP	69.8	73.1	60.6	69.2	**74.2**	**74.2**	65.5
Supplement	59.5	60.6	60.8	**66.5**	66.2	63.7	60.3
GLUE	50.7	52.1	48.9	51.9	69.0	**77.1**	76.6
EWoK	50.7	52.1	47.4	51.9	51.8	**54.3**	51.6
Average	57.7	59.5	54.4	59.9	65.3	**67.3**	63.5

Table 7: Final results compared to the baselines.

learn a representation that balances approximating the perplexity with optimizing for whatever the external objective requires. Thus, it is no surprise that the scores for BLiMP and EWoK are lower. This does not necessarily mean that this model is less capable of understanding grammaticality, but this could not be captured by BLiMP. We are not aware of another benchmark that would resolve this issue.

4.4 Submission

In Table 7, we show the overall results for our best models, compared to the baselines. The results from BabyLlama and LTG-BERT are taken from the reported scores from the organizers. The "BabyLM" model is our internal baseline, using the same parameters and training as our other models, but trained on the data provided by the organizers. "Half / Half " is a model trained on a mixture of the provided data and contrastive data, and "Contr." is trained on exclusively contrastive data.

As we can see, our models outperform even the provided models trained on 100M overall. We suspect this is for the same reason as we found last year in Edman and Bylinina (2023), where the models trained on too large of a context size have trouble converging. In terms of the data used, we see that using the contrastive dataset hurts BLiMP performance, but raises GLUE performance. Using a mix is able to capture a best of both worlds, retaining performance on BLiMP while even improving performance on GLUE and EWoK.

5 Conclusion

In this year's BabyLM Challenge, we attempted to buck the trend of administering strategies based on L1 acquisition, having seen little success from such strategies in last year's Challenge. Instead, we hypothesized that L2 acquisition, with more explicit information regarding semantics and syntax, might be what a language model needs. To that end, we also saw limited success. Our strategy of using Wiktionary data did not show any indication of improved output quality. Using grammar information did have a small positive effect on BLiMP scores, though it is unclear whether the grammar itself helped or simply the more diverse data domain.

Nevertheless, our strategy of reducing context size from the previous year was yet again successful at outperforming the baselines, even those with 10× more data used in training. Additionally, using data that includes paraphrases and contrastive pairs helped improve the GLUE scores by a remarkable 8 points. This goes to show that the data chosen for low-resource pretraining can have a profound impact. The study of the exact structure of data that LMs efficiently learn from is a productive future direction, as tentatively shown by our results.

Acknowledgements

We thank BabyLM anonymous reviewers for useful comments. We also thank Oleg Serikov for helpful informal discussions. The work was supported by the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No. 101113091) and by the German Research Foundation (DFG; grant FR 2829/7-1).

References

Asmik Brutjan and Karine Brutjan. 2022. *Learn English with short stories. A Textbook with Grammar References for Pre-intermediate and Intermediate Learners.* A. Brutjan.

Lukas Edman and Lisa Bylinina. 2023. Too much information: Keeping training simple for BabyLMs. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 89–97, Singapore. Association for Computational Linguistics.

Sidney Greenbaum and Gerald Nelson. 2009. *An Introduction to English Grammar*. Pearson Education.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mark Newson. 2006. *Basic English Syntax with Exercises*. Bölcsész Konzorcium.

Eric H Roth and Toni Aberson. 2010. *Compelling Conversations: Questions and Quotations on Timeless Topics: An Engaging ESL Textbook for Advanced Students*. Chimayo Press.

Audrey Jean Thomson and Agnes V Martinet. 2015. *A Practical English Grammar*. New York: Oxford University Press.

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 Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 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, Aaron Mueller, Leshem Choshen, Ethan Gotlieb Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Adina Williams, Bhargavi Paranjabe, Tal Linzen, and Ryan Cotterell. 2023. Findings of the 2023 BabyLM Challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the 2023 BabyLM Challenge*. Association for Computational Linguistics (ACL).

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.

Laura Wright. 2024. *English Grammar for Literature Students: How to Analyse Literary Texts*. De Gruyter Mouton.

Junlei Zhang, Zhenzhong Lan, and Junxian He. 2021. Contrastive learning of sentence embeddings from scratch. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Less is More: Pre-Training Cross-Lingual Small-Scale Language Models with Cognitively-Plausible Curriculum Learning Strategies

Suchir Salhan 🍋🍊 **Richard Diehl Martinez** 🍋 **Zébulon Goriely** 🍋 **Paula Buttery** 🍋🍊
🍋 Department of Computer Science & Technology, University of Cambridge, U.K.
🍊 ALTA Institute, University of Cambridge, U.K.
{sas245,rd654,zg258,pjb48}@cam.ac.uk

Abstract

Curriculum Learning has been a popular strategy to improve the cognitive plausibility of Small-Scale Language Models (SSLMs) in the BabyLM Challenge. However, it has not led to considerable improvements over non-curriculum models. We assess whether theoretical linguistic acquisition theories can be used to specify more fine-grained curriculum learning strategies, creating age-ordered corpora of Child-Directed Speech for four typologically distant language families to implement SSLMs and acquisition-inspired curricula cross-lingually. Comparing the success of three objective curricula (GROWING, INWARDS and MMM) that precisely replicate the predictions of acquisition theories on a standard SSLM architecture, we find fine-grained acquisition-inspired curricula can outperform non-curriculum baselines and performance benefits of curricula strategies in SSLMs can be derived by specifying fine-grained language-specific curricula that precisely replicate language acquisition theories.

https://github.com/suchirsalhan/MAO-CLIMB (CC BY 4.0)

🤗 https://huggingface.co/climb-mao (CC BY 4.0)

1 Introduction

Curriculum Learning (CL) has emerged as a promising method to improve the cognitive plausibility of **Small-Scale Language Models (SSLMs)** in the first BabyLM Challenge (Warstadt et al., 2023), as a way to gradually introduce more complex linguistic phenomena into the model later in training in a manner that is similar to human language acquisition. Cognitively-inspired SSLMs are models trained on corpora that approximate the volume and nature of input that a first-language learner can expect to receive during language acquisition. These have been found to perform competitively against LLMs in English (Huebner et al., 2021). CL strategies implemented in the BabyLM Challenge either specified a static measure of linguistic complexity, such as lexical frequency (Borazjanizadeh, 2023), sorted datasets according to difficulty (Opper et al., 2023), or gradually increased vocabulary sizes (Edman and Bylinina, 2023). While the majority of these strategies did not yield consistent improvements over non-curriculum learning baselines (Warstadt et al., 2023), linguistic theory suggests that children naturally focus on input that is neither too simple nor too difficult but at the right level of challenge for learning (Biberauer, 2019; Bosch, 2023). This is known as the "Goldilocks Effect", which is a form of self-selecting curriculum learning that appears to naturally occur in first language (L1) acquisition. This raises the question of whether acquisition theories can provide insights into more effective curriculum learning strategies for SSLMs, and lead to more consistent benefits of CL strategies.

Our work assesses whether language acquisition theories can provide us with better heuristics for good curriculum learning strategies to train SSLMs. We compare contrastive acquisition theories for their success when informing objective curriculum learning strategies on a standard architecture (Diehl Martinez et al., 2023). We train SSLMs with three new objective curricula called GROWING, INWARDS and MMM, each replicating the developmental sequences of contemporary acquisition theories that first-language monolingual learners are theorised to follow in the earliest stages of acquisition cross-linguistically. In practice, these curricula modify the standard masked language modelling objective in BabyBERTa-style models by varying the order and the sequence of masking using different tagsets to simulate different language acquisition theories.

The acquisition models specify different cross-lingual and language-specific developmental sequences that learners appear to follow in first language acquisition, which has not been implemented

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 174–188
November 15-16, 2024

or evaluated in the context of Deep Learning. The multilingual focus of the acquisition models is a goal strongly aligned with the spirit of the BabyLM Shared Task. We train SSLMs with these objective curricula for four typologically distant language families: Romance (French), Germanic (German), Japonic (Japanese) and Sino-Tibetan (Chinese). We introduce new age-ordered corpora of Child-Directed Speech (CDS) for these languages and select languages for pre-training based on the quantity of CDS that can be used to train SSLMs using similar volumes of data that learners can utilise in first language acquisition. We evaluate these SSLMs on syntactic minimal pair datasets. We find benefits of the cognitively-inspired objective curricula cross-linguistically, however different strategies lead to better performance for certain languages, particularly finer-grained language-specific versions of the MMM objective. Acquisition-inspired objective curricula can obtain comparable performance on minimal pair evaluation datasets to LLMs, despite requiring approximately 25X fewer parameters and 6,000X fewer words.

2 Background

We survey Curriculum Learning (CL) strategies used in the 1st BabyLM Challenge *Section* 2.1 and contrastive models of syntactic acquisition that are utilised to replicate cross-lingual developmental sequences for implementing more cognitively plausible pre-training in SSLMs in *Section* 2.2.

2.1 Curriculum Learning Strategies for Pre-training on Developmentally Plausible Corpora

While some SSLMs that utilised CL strategies outperformed the official BabyLM baselines, no CL strategies led to consistent or uniform improvements compared to stronger non-curriculum models. Many submissions for the inaugural BabyLM Challenge utilised Curriculum Learning on a small-scale masked language model architecture trained on a 5 million (5M) word corpus called BabyBERTa (Huebner et al., 2021), based on a Transformer Language Model RoBERTa (Liu et al., 2019) with 15$\times$ fewer parameters, which displayed comparable grammatical capabilities to RoBERTa. In general, CL strategies, like using a pre-defined static difficulty assessment based on linguistic criteria like syntax dependency tree depth (Oba et al., 2023) or ranking sentences according to surprisal (Chobey et al., 2023) or length (DeBenedetto, 2023) or other measures of difficulty (Opper et al., 2023), showed little improvement over non-CL baselines. Diehl Martinez et al. (2023) introduce **Curriculum Learning for Infant-Inspired Model Building (CLIMB)**, which incorporates three CL strategies into BabyBERTa pre-training that each dynamically increase the difficulty of the language modelling task throughout training. CLIMB's **vocabulary curriculum** constrains the Transformer vocabulary in the initial stages of training by dynamically mask out vocabulary units over training. CLIMB's **data curriculum** varies the order of training instances based on infant-inspired expectations and the learning behaviour of the model, enabling dynamic sampling of training data according to a difficulty function. CLIMB's **objective curriculum** combines the masked language modelling task, used in RoBERTa (Liu et al., 2019) and the BabyBERTa model (Huebner et al., 2021), with coarse-grained word class prediction to reinforce linguistic generalisation capabilities. This provides functionality to change the objective function at specified discrete training steps. The objective curricula modifies the Masked Language Modelling (MLM) objective, which is the standard "denoising" objective for Pre-trained Language Models, like RoBERTa and BabyBERTa. Both models use a random token masking strategy, applying a fixed masking ratio α to mask different contexts selected randomly with a probability P_i. Diehl Martinez et al. (2023) introduce two objective curricula defined using 'curriculum units' of Universal Part of Speech (UPOS) tags. The first objective classifies [MASK] to one of [Verb, Noun, Other], while the second objective classifies [MASK] to one of the 10 UPOS tags. CLIMB's objective curricula, following the submission guidelines of the 1st BabyLM Challenge, are performed using an unsupervised part-of-speech (POS) tagger. They additionally tuned the vocabulary and model size of BabyBERTa, resulting in a model that outperformed the official baselines for the first BabyLM Challenge. CLIMB's curriculum learning strategies outperformed the official baseline but the accuracy of CL-strategies was comparable to the stronger BabyBERTa-style baseline introduced by the authors. We add new **cognitively-plausible objective curricula**, as an extension to the original CLIMB submission and CLIMB's improved BabyBERTa-style as baselines.

2.2 Acquisition Models in Deep Learning: Three Models

To assess whether using acquisition theories can be used to formulate better-performing CL strategies, we consider three recent language acquisition models that are amenable to Deep Learning implementation, as they specify developmental sequences that can be replicated as CL strategies in SSLMs. Based on careful linguistic analysis of universal and language-specific patterns in the utterances produced by learners cross-linguistically at different stages of acquisition, linguists have formalised strict (universal or non-language-specific) or weak (language-specific) orders of syntactic categories that are sequentially acquired. Since these acquisition models have been formulated based on linguistic analysis of multilingual acquisition data, we consider whether the CL strategies that precisely replicate these models can inform better-performing curriculum learning strategies cross-lingually. This leads us to train SSLMs with these objective curricula beyond English. As schematised in *Figure* 1, we can precisely replicate these developmental sequences as stages of SSLM pre-training, defined as proportions of training steps.

We implement three contemporary cross-lingual models of syntactic acquisition:

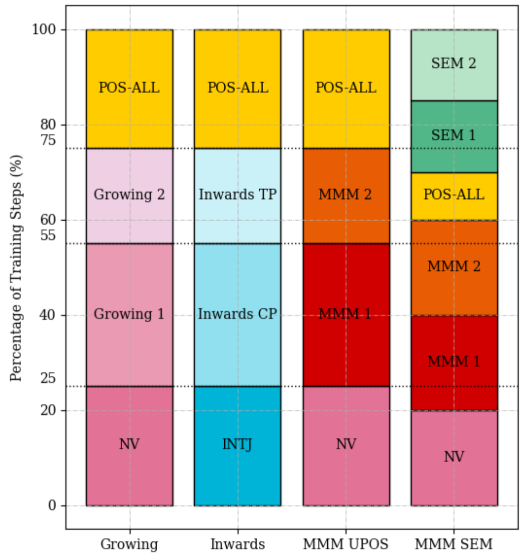


Figure 1: **Acquisition-inspired Objective Curricula:** We specify Objective Curricula GROWING, INWARDS, MMM (UPOS), MMM (SEMANTIC) for three theories of acquisition (*Section* 2.2). The Progression of Curriculum Units replicate the predicted developmental sequences by specifying curriculum units (defined in *Table* 1) defined over different pre-training stages, expressed as a percentage of training steps.

1. **GROWING:** Bottom-up maturational approaches to language acquisition (Rizzi, 1993; Radford, 1990), including the "Growing Trees Hypothesis"(Friedmann et al., 2021), predicts that first language learners begin acquiring verbs and nouns (unit NV in *Table* 1). Learners subsequently progress to acquiring predicate information to form simple sentences; and finally, acquire discourse and complementiser information, allowing them to formulate complex sentences (e.g., with relative clauses). We can assume a tripartite model of bottom-up maturational development for implementation, with units Growing 1 and Growing 2 in *Table* 1.[1]

2. **INWARDS:** Bosch (2023) introduces the predictions of a **generalised inward-growing maturational proposal (INWARDS)**, building on evidence from Heim and Wiltschko (2021) of early acquisition of "discourse"-material and interactional language (e.g. tags-questions). This predicts exactly the opposite order of acquisition of GROWING. The stages of development begin with the early acquisition of complementisers used for illocutionary/discourse-related purposes (INTJ and INWARDS- CP in *Table* 1); followed by the acquisition of tense/event-related information (INWARDS-TP); and finally, thematic information.

3. **NEO-EMERGENT (MMM):** Neo-Emergentism predicts developmental stages in language acquisition that show increasing categorial granularity, taking a language-specific, or non-maturational, approach towards syntactic acquisition (Biberauer and Roberts, 2015). The general universal prediction of one neo-emergent model called Maximise Minimal Means (MMM) is that all learners, irrespective of

[1] There are differences in the number of stages predicted in bottom-up maturational approaches. Bottom-up approaches (Rizzi, 1993; Radford, 1990) predict tripartite developmental sequence (a Verb Phrase, Tense Phrase and Complementiser Phrase), but Growing Trees involves bipartite stages (TP and VP is Stage 1, and Stage 2 involves acquiring the CP until QP to predict early acquisition of WH-questions).

the language being acquired, follow the same "coarse" stages in the acquisition of syntactic categories. They first learn to distinguish nouns and verbs (Unit NV), and then an "intermediate" set of categories (complementisers and event-related words),[2] before finally learning tense/aspectual categories (units MMM 1 and MMM 2 in *Table* 1). We implement this as a **universal "coarse" default curriculum strategy** that we implement as a default curriculum strategy (MMM (UPOS) in *Figure* 1). However, MMM also incorporates **language-specific differences in "finer-grained" curricula** where learners can acquire language-specific categories, leading to typological variation in the order of acquisition (Biberauer, 2019; Bosch, 2023, 2024), which we try to model in a CL strategy by specifying language-specific tagsets in SEM 1, SEM 2 in *Table* 1.

Unit	POS Tags
NV	[NOUN, VERB]
Growing 1	NV+ [DET, ADJ, PRON, PROPN, NUM, PRT]
Growing 2	growing_1 + [AUX, PART, ADP, ADV]
INTJ	[X, INTJ, SYM]
INWARDS CP	INTJ+ [PROPN, CCONJ, SCONJ, SYM]
INWARDS TP	CP+ [NUM, PRT, AUX PART, ADP, ADV]
MMM 1	NV+ [DET, CONJ, INTJ]
MMM 2	MMM 1 + [ADJ, ADV, PRON, PROPN, NUM, PRT]
SEM 1	UPOS $+t_{\text{sem}} \in$ [EVE, TNS, ACT, ANA]
SEM 2	SEM1 + $+t_{\text{SEM}} \in$[LOG, COM, DEM, DIS, MOD, ENT, NAM, TIM]

Table 1: Summary of Curriculum Units comprise Universal Part-of-Speech Tags and the Semantic Tags introduced by Bjerva et al. (2016) used to define GROWING, INWARDS & MMM objective curricula. The ordering of units for each acquisition-inspired curriculum is shown in *Figure* 1.

Each stage of the GROWING, INWARDS and MMM models can be defined as a 'curriculum unit' composed of POS tag sequences listed in *Table* 1.[3] To precisely replicate the developmental sequences of each acquisition model computationally, we will need to use a supervised tagger to specify curricula using strictly ordered sequences of POS tags. This is a cognitively motivated divergence from Diehl Martinez et al. (2023), who use an unsupervised tagger to define curricula. Using a supervised tagger is argued by Buttery (2006) to enable computational modelling of a more cognitively plausible starting point for first language (L1) learners – based on a view of acquisition that is not fully emergent, nor completely nativist.[4] For our purposes, it allows us to precisely replicate developmental sequences in SSLMs using curriculum learning.

3 Dataset

3.1 Training Corpora: MAO-CHILDES

We collect a training corpus of Age-ordered Child-Directed Speech (CDS) for four languages (French, German, Japanese and Chinese), in addition to the English Age-Ordered-CHILDES (AO-CHILDES) corpus (Huebner and Willits, 2021) used in the BabyLM Challenge, to assess the benefits of the acquisition-inspired curricula beyond English compared to non-curriculum SSLMs. MAO-CHILDES is developed from the Child Language Data Exchange System (CHILDES) (MacWhinney, 2000), which consists of in-home recordings of casual speech from caregivers to children and in-lab activities such as play, conversation and book reading directed towards first language learners for several languages.[5] We make our training corpus available on HuggingFace.[6] The distribution of CHILDES data beyond English is a practical challenge for extending the BabyLM Challenge beyond English. *Table* 6 shows the imbalance in quantities of CDS extracted from CHILDES, which is an artefact of a Western, Educated, Industrialised, Rich, and Democratic (WEIRD) bias in language acquisition research (Henrich et al., 2010). A sample of CDS in the age-ordered corpora is shown in *Figure* 2, from different stages of language acquisition. Following Huebner and Willits (2021), utterances

[2]In Chomskyan terminology, a vP-shell and a Complementiser Phrase (CP).

[3]The Chomskyan acquisition models used in this paper technically refer to syntactic projections, rather than part-of-speech tags.

[4]Note that Buttery (2006) uses a model within a Combinatorial Categorial Grammar (CCG)-based formalism, which is also a "middle ground" between fully emergent acquisition models and a traditional biologically hardwired Universal Grammar assumed in traditional Chomskyan models like Principles and Parameters.

[5]Original data can be accessed here: https://childes.talkbank.org/

[6]https://huggingface.co/climb-mao

from children and child-directed speech (CDS) produced by caregivers, and other interlocuters, to children over the age of 6;0 are disregarded, leaving CDS produced by caregivers to children less than 6;0 which is sorted using the meta-data of the age of the learner in the CHILDES database.[7]

où tu vas?
Where are you going?
où_PRON tu_VERB vas_NOUN

Stage 1
MLU 1.3 (range 1.09 – 1.57; average length of 3.4 months).

je le racle et après je te le donne
I scrape it and give it to you.
je_PRON le_DET racle_NOUN et_CCONJ après_ADP je_PRON te_VERB le_PRON donne_VERB

Stage 2
MLU 1.69 (range 1.44–1.96; average length of 7.8 months).

ils ne cueillent pas quelque chose
They don't pick something
ils_PRON ne_ADV cueillent_VERB pas_ADV quelque_DET chose_NOUN

Stage 3
MLU 2.82 (range 2.32–3.57).

Figure 2: A sample of Child-Directed Speech (CDS) from French **MAO-CHILDES** that learners receive from caregivers at different stages of acquisition. Stages of acquisition are standardly defined in terms of mean lengths of utterances produced by learners.

3.2 Evaluation Datasets

To assess the success of three objective curricula (GROWING, INWARDS and MMM) that precisely replicate the predictions of the acquisition theories in *Section* 2.2 on a standard SSLM architecture in a multilingual setting, we extend the evaluation pipeline of the BabyLM Challenge. This consists of syntactic evaluation datasets like BLiMP (Warstadt et al., 2020) composed of minimal pairs of grammatical and ungrammatical sentences for language-specific syntactic phenomena. We use the following minimal pairs datasets to evaluate the objective curricula for the four languages in MAO-CHILDES:

1. **CLAMS (French and German):** The Cross-Lingual Syntactic Evaluation of Word Prediction Models (CLAMS) (Mueller et al., 2020) generates minimal pair datasets which we use for French and German using Attribute-Varying Grammars. The dataset assesses grammaticality in Simple Agreement, VP coordination, and across "interveners" in S-V agreement (subject/object relative clause or across a Prepositional Phrase).
2. **JBLIMP (Japanese):** JBLIMP (Someya and Oseki, 2023) is a minimal pairs dataset for targeted syntactic evaluation of Japanese. It consists of 331 minimal pairs of syntactic acceptability judgements curated from Japanese syntax articles in the *Journal of East Asian Linguistics*.[8]
3. **SLING (Chinese): SLING** (Song et al., 2022) is a 38K minimal sentence pair dataset derived by applying syntactic and lexical transformations to Chinese Treebank 9.0,[9] aiming to improve on the limitations of an earlier dataset called CLiMP (Xiang et al., 2021), which had a lack of diversity in the vocabulary to generate minimal pair templates.

Due to the small size of the JBLIMP minimal pairs dataset, we follow Someya and Oseki (2023)'s recommendation to compute accuracy using a SLOR score to mitigate the confounding effects of lexical frequencies and sentence lengths, which is defined as follows:

$$SLOR(X) = \frac{\log p_m(X) - \log p_u(X)}{|X|}$$

where $p_m(X)$ is the probability of a sentence for a Language Model and is the unigram probability of the sentence, estimated for each subword in the training corpus. Accuracy calculations for other languages follows dataset guidance to use unnormalised log-probabilities.

3.3 Universal POS Tagging

To define fine-grained objective curricula that perform masked language modelling with different subsets of syntactic and semantic tags for a specified proportion of training steps, we have to annotate child-directed speech corpora with Universal POS tags using an off-the-shelf SpaCy multilingual POS tagger. The distribution of POS tags in MAO-CHILDES (Figure 4) contains a high proportion of Nouns, whereas Verbs contribute a relatively low count. There are orthographic issues in the CHILDES dataset for East Asian Languages,

[7] The Script for Generating AO-CHILDES can be found here:https://github.com/UIUCLearningLanguageLab/AOCHILDES

[8] The JBLiMP Minimal Pair dataset can be found here: https://github.com/osekilab/JBLiMP/tree/main

[9] The SLING Dataset can be found here: https://huggingface.co/datasets/suchirsalhan/SLING

which are transcribed using Romanised characters (romaji) and a large proportion of English loan words in the Japanese portion of MAO-CHILDES, used in certain lexical domains, are incorrectly tagged automatically. These pre-processing inconsistencies were manually corrected. We also train a semantic tagger to specify language-specific curriculum strategies (see *Appendix* A for more detail).

4 Methodology

4.1 Model Architecture

Following Diehl Martinez et al. (2023), we develop non-curriculum learning models. These models are scaled-down language models based on RoBERTa (Liu et al., 2019), with 8M parameters and trained on no more than 30M words (Huebner et al., 2021). We use 8192 vocabulary items, which Diehl Martinez et al. (2023) find yields better overall performance compared to a larger vocabulary. Token unmasking is also removed, like BabyBERTa. We use a small model architecture composed of eight layers. This follows Diehl Martinez et al. (2023), who compare the role of model size $(8, 10, 12$ Transformer layers) and vocabulary size (comparing $|V| \in \{8192, 16384\}$). An AdamW optimiser with linear scheduling is used (Loshchilov et al., 2017). Each model is trained for 400,000 steps with 4 A100 GPUs. The hyperparameters used for the "vanilla" SSLMs are shown in *Table* 4. The models concatenate input sequences to capitalise on the available input length.

4.2 Baselines: LLMs and SSLM (WIKI)

We use two families of models as baselines. First, we compare the performance of monolingual SSLMs to monolingual Large Language Models to assess the benefits of the BabyLM paradigm. For French, German and Chinese, we use RoBERTa-style monolingual LLMs.[10] The Chinese RoBERTa model is trained on around 30B words (Cui et al., 2020), which more than 10^4 times the training data we use to train our SSLMs in the Chinese portion of MAO-CHILDES.[11] We include GPT-2 Baselines for Japanese, which are reported by Someya and Oseki (2023). This is because Japanese RoBERTa monolingual language models[12] are not trained on data using Romaji orthography, which is used in the Japanese portion of MAO-CHILDES (*Section* 3). Secondly, to assess the benefits of pre-training SSLMs on Child-Directed Speech, we train SSLMs using Wikipedia text (SSLM WIKI), which is extracted to match the quantity of training data in MAO-CHILDES for each language. We keep the original hyperparameter settings used by Huebner et al. (2021).

4.3 "Vanilla" SSLMs: MAO-BabyBERTa

We train a family of SSLMs, called Monolingual Age-Ordered BabyBERTa (MAO-BABYBERTA), on language-specific training data from MAO-CHILDES using the model architecture described in *Section* 4.1 without any curriculum learning strategies. Hyperparameters are tuned for English, and we use the same settings in MAO-BabyBERTa.

4.4 Implementing Acquisition-Inspired Objective Curricula: GROWING, INWARDS & MMM

To implement the acquisition-inspired strategies, we filter our age-ordered MAO-CHILDES corpus for each language for expected utility in the acquisition process, according to the curriculum strategies of GROWING, INWARDS and MMM schematised in *Figure* 1. We then precisely implement the GROWING, INWARDS, MMM theories introduced in *Section* 2.2, using different curriculum units composed of POS tagsets (*Table* 1) to define three objective curricula that replicate the developmental sequences of each acquisition model through the progressive ordering of POS units. The logic for performing masked language modelling selectively for words annotated with a desired set of specified part of speech tags is implemented in Diehl Martinez et al. (2023), which we extend. The objective curricula modify the masked language modelling (MLM) objective in a multi-task learning setup, so the acquisition-inspired objective is activated and optimised in parallel with MLM. We fix the model architecture to be identical to the "vanilla" SSLM architecture in *Section* 4.3 to evaluate the benefits of each curriculum strategy. We modify CLIMB's objective curricula to implement the GROWING, INWARDS and MMM objective curricula by splitting 400K training steps across

[10]The French RoBERTa model is available here: https://huggingface.co/abhilash1910/french-roberta. The German RoBERTa model is available here: https://huggingface.co/uklfr/gottbert-base

[11]The Chinese RoBERTa model is available here: https://huggingface.co/hfl/chinese-roberta-wwm-ext-large.

[12]Japanese RoBERTa models is available here: https://huggingface.co/rinna/japanese-roberta-base

	Model		English	Japanese	Chinese	French	German
Non-CL	SSLM (WIKI)		64.60%	55.42%	48.01%	70.68%	59.63%
	MAO-BABYBERTA		75.48% *	61.21%	51.32%	**80.00%**	68.78%
CL	GROWING		71.13%	79.30%	56.22%	76.21%	71.13%
	INWARDS		71.05%	81.32%	54.26%	79.01%	69.34%
	MMM	(UPOS)	74.22%	**87.31%**	**58.79%**,	75.93%	**73.25%**
		(SEM)	**77.35%**		55.01%		

Table 2: Evaluation of MAO-BABYBERTA ("vanilla" SSLM architecture without objective curricula) and the three Objective Curricula (GROWING, INWARDS, and MMM) on the following syntactic minimal pairs datasets: BLIMP (English), JBLIMP (Japanese), SLING (Chinese), CLAMS (French and German). Performance is compared to SSLM (WIKI). This is the same architecture trained on non-CDS training data. *This reports the performance of the best-performing "vanilla" model by Diehl Martinez et al. (2023) on the same architecture used to train our model. **Bolded** results indicate the highest accuracy of all the models.

four non-uniform intervals that are defined as a proportion of the SSLM's training steps, defined in *Figure* 1. This is meant to roughly simulate four developmental stages of an idealised monolingual learner until 6;0. We then specify tagsets for each phase of the curricula that correspond to the acquisition theory. To illustrate this, the INWARDS curriculum begins with a unit INTJ, which performs MLM for interjections and other interactional language, which are annotated with tags INTJ, X, SYM. Then, we specify two further curriculum units INWARDS-CP which performs MLM on complementiser-like words (e.g., SCONJ), and INWARDS-TP which performs MLM on auxiliaries AUX and other tense/event-related words. At each stage of the curriculum, the objective curricula provide the vanilla SSLM model with a list of syntactic tags to use during training, taken from a pre-specified set of UPOS tags that lists all the tags used in the UPOS tagged MAO-CHILDES training set. If a tag is not used at the curriculum stage, its "ID" is set to zero so it is not a target for masked language modelling (MLM). During training, the number of part-of-speech tags that the model has to classify over are varied, according to the predictions of each acquisition model. The objective curricula end with a final curriculum unit, Pos-ALL, containing the entire Universal Part-of-Speech Tagset. The masking ratio is an important hyperparameter that impacts the pretraining of a Masked Language Model. A masking ratio of 0.4 is used for the tags specified at the curriculum stage. A 0.15 masking rate is used elsewhere if the tag is not specified at the curriculum stage. For RoBERTa-based Language Models, a masking ratio of 0.4 performs better than 0.15 in downstream tasks (Wettig et al., 2023).In addition to our "default" MMM strategy defined by Universal POS tags, MMM (UPOS), we additionally introduce a **refined version of the MMM objective**, MMM (SEM) for English and Chinese. This adds two additional stages to the non-language specific strategy to define a language-specific curricula that utilises semantic tags (Bjerva et al., 2016), or *sem*-tags, to model **language-specific acquisition strategies** (Section 2.2). Detailed methods and results are discussed in *Appendix* A. Training times for each objective are summarised in *Table* 5.

5 Results

The performance of objective curricula and cross-lingual SSLMs on minimal pairs datasets is summarised in *Table* 2. **Fine-grained objective curricula demonstrate variable effectiveness compared to non-curriculum baselines**. While MMM (UPOS) shows general promise, average benefits of MMM (UPOS), GROWING, and INWARDS, do not show statistically significant improvements on MAO-BABYBERTA cross-linguistically ($p < 0.05$). However, **the MMM (SEM) curriculum achieves a statistically significant performance improvement in both English and Chinese** ($p < 0.05$) when performing a paired t-test. Instead, **statistically significant improvements are observed with acquisition-inspired CL strategies in specific languages across minimal pairs test sets**. MMM (UPOS) only achieves a statistically significant improvement in Japanese and Chinese. GROWING leads to a statistically significant improvement in Japanese and Chinese, while INWARDS only has statistically significant improvements in Japanese. No curriculum strategy outperforms MAO-BABYBERTA in French, although INWARDS almost reaches the same accuracy. German CL strategies only marginally outperform the non-CL baseline. In *Figure* 3, we compare these results with a broader range of models introduced by Diehl Martinez et al. (2023), finding that the English MMM (SEM) curriculum marginally outperforms other curriculum learning strategies. See *Appendix* C for details on how t-test statistics are computed.

Language	LLM	SSLM (CL)
English	**80.10**	77.35(MMM SEM)
Japanese	77.95	**87.31 (MMM)**
Chinese	**83.41**	58.79 (MMM)
French	**83.00**	79.01(Inwards)
German	**92.16**	73.25(MMM)

Table 3: Comparison of Accuracy of LLMs and the Best Performing CL Strategy on Minimal Pairs Datasets. SEM represents Language-Specific strategies implemented for English and Chinese pre-training compared to the language-invariant MMM (UPOS) strategies.

6 Discussion

Acquisition-inspired CL strategies represent a novel large-scale application of language acquisition theory in Deep Learning, aimed at improving the performance of SSLMs. Acquisition-inspired curricula guide SSLMs, which function as large statistical learners, to generalise over frequent linguistic categories—such as nouns and verbs—early in the training process and attend to language-specific features, such as the Germanic V2 word order. This suggests that **more fine-grained, language-specific curricula may have performance benefits over non-CL strategies in SSLMs**, which is supported by results showing the limited improvements of universal/maturational theories of acquisition that inform the GROWING and INWARDS strategies. Although both acquisition models predict universal curricula that should lead to consistent benefits cross-lingually, GROWING/INWARDS only improve performance in Chinese and Japanese, while performing comparably to non-curriculum (non-CL) baselines in French/German and worse than non-CL baselines in English. An additional benefit of using fine-grained language-specific curricula is that it enables SSLMs to learn more complex grammatical phenomena that may rely on semantics like anaphora. We notice notable improvements in ellipsis performance (*Table* 7) with the MMM (SEM) curriculum. Interestingly, in Chinese, the MMM (SEM) curriculum marginally underperforms compared to MMM (UPOS) when handling anaphora and aspectual phenomena (*Table* 8), highlighting the need for further investigation into engineering optimal language-specific curriculum strategies that outperform non-CL strategies. This raises important avenues for future research. Careful analysis of developmental sequences beyond English to develop language-specific strategies similar to MMM (UPOS/SEM) will be crucial. We encourage practitioners to curate larger corpora of child-directed speech (CDS) for training SSLMs in languages beyond English and to develop more minimal pair datasets that have coverage beyond grammatical agreement in CLAMs to develop better-performing curriculum strategies for Romance and Germanic. Additionally, an important finding is that **acquisition-inspired CL strategies in Japanese significantly outperform GPT-2** (*Table* 3). The improvements observed in Japanese control/raising phenomena (*Table* 9) suggest that the properties of CDS in Japanese may lead to more robust generalisations than LLMs.

7 Conclusion

This paper assesses whether fine-grained curriculum learning strategies based on acquisition theories can provide better heuristics for CL strategies for SSLM pre-training cross-lingually, introducing the MAO-CHILDES training corpus to train SSLMs for four typologically distant language families. Mixed results of the maturational GROWING and INWARDS acquisition theories in curriculum strategies and the implementation of the coarse/universal prediction of MMM (UPOS) suggest that there is no guaranteed performance benefit just by devising universal CL strategies based on acquisition theories for SSLMs in a multilingual setting. Training SSLMs using more fine-grained language-specific curricula that precisely replicate cutting-edge linguistic theories is effective for the MMM (SEM) objective in English and Chinese and MMM (UPOS) in Japanese. Curriculum Learning can outperform non-curriculum SSLMs by specifying fine-grained language-specific curricula that precisely replicate language acquisition theories, highlighting how cognitively-inspired techniques can lead to better-performing data-efficient architectures in the spirit of the BabyLM Challenge.

Acknowledgments

Many thanks to Andrew Caines for his comments, supervision and feedback on this paper. We thank Núria Bosch-Masip for her comments on the linguistic acquisition models implemented in this paper. We thank Mila Marcheva for her thoughts on cognitively-inspired modelling, which influenced the ideas in this paper. This paper reports on work supported by Cambridge University Press & Assessment. It was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council. Additionally, we thank the NVIDIA Corporation for the donation of the Titan X Pascal GPU used in this research. Richard Diehl Martinez is supported by the Gates Cambridge Trust (grant OPP1144 from the Bill & Melinda Gates Foundation). Zébulon Goriely's work is supported by The Cambridge Trust.

References

Theresa Biberauer. 2019. Children always go beyond the input: The Maximise Minimal Means perspective. *Theoretical Linguistics*, 45(3-4):211–224.

Theresa Biberauer and Ian Roberts. 2015. Rethinking Formal Hierarchies: A Proposed Unification. *Cambridge Occasional Papers in Linguistics*, 7:1–31.

Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3531–3541, Osaka, Japan. The COLING 2016 Organizing Committee.

Nasim Borazjanizadeh. 2023. Optimizing GPT-2 pretraining on BabyLM corpus with difficulty-based sentence reordering. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 356–365, Singapore. Association for Computational Linguistics.

Núria Bosch. 2024. On another topic, how do acquisition orders vary? The left periphery and topicalisation in bilinguals and monolinguals. 1st year PhD report.

Núria Bosch. 2023. Emergent Syntax and Maturation: A Neo-Emergentist Approach to Development. *MPhil Thesis, Department of Theoretical and Applied Linguistics, University of Cambridge*.

Paula J. Buttery. 2006. Computational models for first language acquisition. Technical Report UCAM-CL-TR-675, University of Cambridge, Computer Laboratory.

Aryaman Chobey, Oliver Smith, Anzi Wang, and Grusha Prasad. 2023. Can training neural language models on a curriculum with developmentally plausible data improve alignment with human reading behavior? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 98–111, Singapore. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Justin DeBenedetto. 2023. Byte-ranked curriculum learning for BabyLM strict-small shared task 2023. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 198–206, Singapore. Association for Computational Linguistics.

Richard Diehl Martinez, Hope McGovern, Zebulon Goriely, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – Curriculum Learning for Infant-inspired Model Building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 112–127, Singapore. Association for Computational Linguistics.

Lukas Edman and Lisa Bylinina. 2023. Too much information: Keeping training simple for BabyLMs. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 89–97, Singapore. Association for Computational Linguistics.

Naama Friedmann, Adriana Belletti, and Luigi Rizzi. 2021. Growing trees: The acquisition of the left periphery. *Glossa: a journal of general linguistics*, 6(1):131.

Jutta Heim and Martina Wiltschko. 2021. Acquiring the form and function of interaction: a comparison of the acquisition of sentence-final particles and tag questions in the brown corpus. Talk presented at LAGB Annual Meeting 2021 (online), 8 September.

Joseph Henrich, Steven J Heine, and Ara Norenzayan. 2010. The weirdest people in the world? *Behavioral and Brain Sciences*, 33(2-3):61–83.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Philip A Huebner and Jon A Willits. 2021. Using lexical context to discover the noun category: Younger children have it easier. In *Psychology of learning and motivation*, volume 75, pages 279–331. Elsevier.

Wenxi Li, Yiyang Hou, Yajie Ye, Li Liang, and Weiwei Sun. 2021. Universal semantic tagging for English and Mandarin Chinese. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5554–5566, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov, Frank Hutter, et al. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5.

Brian MacWhinney. 2000. *The CHILDES Project: The Database*, volume 2. Psychology Press.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Linguist.*, 19(2):313–330.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Miyu Oba, Akari Haga, Akiyo Fukatsu, and Yohei Oseki. 2023. BabyLM challenge: Curriculum learning based on sentence complexity approximating language acquisition. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 290–297, Singapore. Association for Computational Linguistics.

Mattia Opper, J. Morrison, and N. Siddharth. 2023. On the effect of curriculum learning with developmental data for grammar acquisition. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 346–355, Singapore. Association for Computational Linguistics.

Andrew Radford. 1990. The Syntax of Nominal Arguments in Early Child English. *Language Acquisition*, 1(3):195–223.

Luigi Rizzi. 1993. Some Notes on Linguistic Theory and Language Development: The case of root infinitives. *Language Acquisition*, 3(4):371–393.

Suchir A. Salhan. 2023. On the potential for 'Maximising Minimal Means' in Transformer Language Models: A Dynamical Systems Theory Perspective. *Cambridge Occasional Papers in Linguistics*, page 55–110.

Taiga Someya and Yohei Oseki. 2023. JBLiMP: Japanese benchmark of linguistic minimal pairs. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1581–1594, Dubrovnik, Croatia. Association for Computational Linguistics.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, and Mohit Iyyer. 2022. SLING: Sino linguistic evaluation of large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4606–4634, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023. Call for papers – The BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus.

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.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2023. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2985–3000, Dubrovnik, Croatia. Association for Computational Linguistics.

Beilei Xiang, Changbing Yang, Yu Li, Alex Warstadt, and Katharina Kann. 2021. CLiMP: A benchmark for Chinese language model evaluation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2784–2790, Online. Association for Computational Linguistics.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207–238.

A MMM (SEM): Specifying Language-Specific Curricula using Semantic Tags

As a first step towards modelling language-specific curricula using curriculum learning, we use Universal Semantic Tagging (*sem-tagging*) (Bjerva et al., 2016). The set of semantic tags can differ cross-lingually. In Chinese, Li et al. (2021) specifies a language-specific semantic tagset, adding and removing tags based on Chinese's semantic and syntactic properties. The fine-grained curriculum in an SSLM set-up aims to circumvent known problems of shortcut learning in LLMs that prevent Transformer-based models from exhibiting robust structural generalisation capabilities that humans exhibit in acquisition (Salhan, 2023).

We perform *sem*-tagging to annotate the BabyLM corpus for English and the Chinese corpus in MAO-CHILDES with a set of language-neutral tags (sem-tags). For English, we only perform *sem*-tagging for the Adult Directed Speech datasets in the BabyLM Challenge dataset: the BNC, Project Gutenberg , OpenSubtitles, QCRI, Wikipedia and Switchboard corpora. This allows us to modify our UPOS curricula for English to specify a more complex curricula to simulate later stages of language acquisition. The first stage of the new MMM curriculum using semantic tags includes tags related to event, `EVE`, tense, `TNS`, and modality `MOD`. These are typically learnt later during acquisition, as part of complex tense sequences of auxiliaries and modal verbs (Biberauer and Roberts, 2015), and allow us to define a **language-specific** sem-tag objective. For Chinese, we sem-tag a corpus of Wikipedia text that contains the same amount of text as the age-ordered CHILDES corpora introduced in *Section* 3.

A.1 Semantic Tagger Accuracy

A multi-objective POS and *sem*-tagger is trained, using a Bidirectional LSTM (BiLSTM) with a Conditional Random Field (CRF) inference layer to train a multi-objective semantic and UPOS tagger for English and Chinese. This is trained on 1100 *sem*-tagged sentences from the Wall Street Journal (WSJ) section of the Penn Treebank (Marcus et al., 1993) and a 1000 *sem*-tagged sentences from Chinese TreeBank (Xue et al., 2005) annotated by Li et al. (2021). The tagger has 91.4% accuracy for Chinese and 94.6% accuracy for English.

B Training

Table 4: Hyperparameter Settings for CLIMB's "vanilla" and curriculum models and MAO-BabyBERTa (CDS)

Layers	8
Heads	8
Hidden	256
$\|V\|$	$8,192$
Layer Norm EPS	1×10^{-5}
Learning Rate	0.001
Optimizer	AdamW
Scheduler Type	Linear
Max Steps	$400,000$
Warm-up Steps	$100,000$

Type	Model	Training Time
MAO-CLIMB	GROWING	11h 51m
	INWARDS	11h 51m
	MMM (UPOS)	11h 46m
	MMM (SEM)	25h 3m
Vanilla Models	CLIMB-small-raw	12h

Table 5: Compute required to train our models. We report the model with the shortest and longest runtime for each experiment type. Each model is trained for 400,000 steps with 4 A100 GPUs.

C Statistical Significance & Detailed Results

The statistical significance of the three curriculum strategies, GROWING, INWARDS & MMM is calculated by performing t-tests on the detailed results in *Tables* 7, 8, 9, 10. For each curriculum (GROWING, INWARDS, MMM (UPOS), MMM (SEM)), we calculate the paired differences in accuracy with the Vanilla model for all the test sets in the minimal pairs evaluation dataset. We perform paired t-tests for the non-CL baseline (MAO-BABYBERTA) and the accuracy of the respective curriculum for each curriculum strategy for each language, concluding that the curriculum-based model significantly outperforms the Vanilla/MaoBabyBERTa model if the p-value is below our significance level $\alpha = 0.05$. The detailed results, below, support the findings of Huebner et al. (2021) cross-linguistically of the benefits of using less training data and paying careful attention to training artefacts and the domain of training corpora, as using CDS to train SSLMs (with/without objective curricula) outperforms SSLM (WIKI).

Figure 3: **Comparision of BLiMP Performance of English SSLMs with CLIMB curricula and GROWING, INWARDS, MMM (UPOS), MMM (SEM) (*Section* 4.4)** We report introduced by Warstadt et al. (2023) for T5-base and OPT-125m models. We include the improved BabyBERTa baseline implemented in Diehl Martinez et al. (2023), which beat the baseline used in the 1st BabyLM Shared Task. We report BLiMP performance of different CLIMB `small-raw` models (also used in the standard architecture of MAO-BABYBERTA used with the three objective curricula) for the best performing dynamic curriculum learning strategies implemented in Diehl Martinez et al. (2023). This includes CLIMB's **Data Curriculum** (Log Pacing with Source Difficulty), **Vocabulary Curriculum** (Log Pacing with Token ID Difficulty), two **Objective Curricula strategies** (MLM + ALL uses a multitask objective of masked language modelling and objective curricula specified by 10 tags throughout all training steps, MLM + NV uses three tags throughout training), and the best performing **Combination Model** (Token ID Vocabulary Curricula, Random + model ppx Data Curricula, Multitask Objective Curricula).

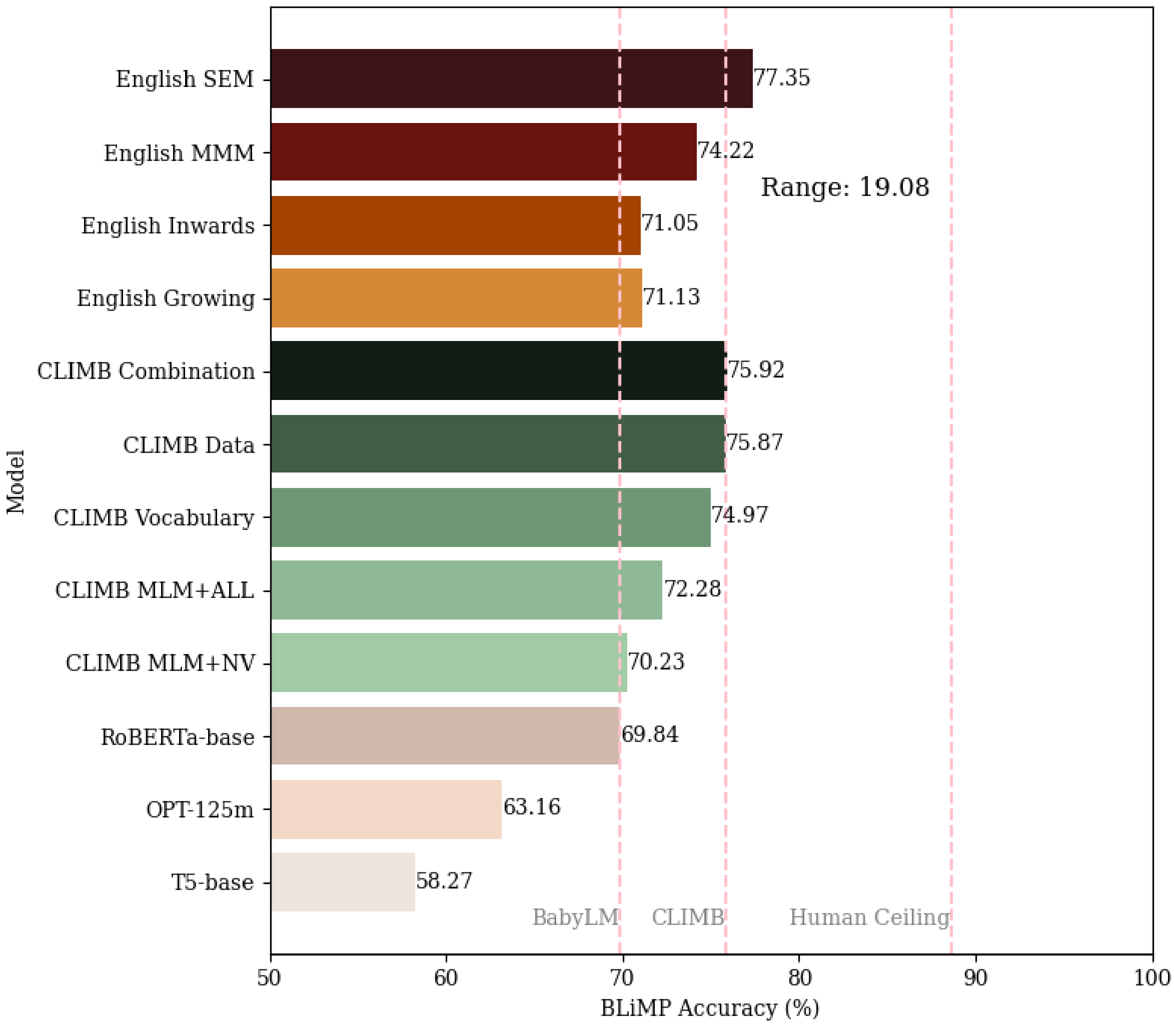


Table 6: Corpus Statistics for the **Child-Directed Speech (CDS)** files extracted from CHILDES for 24 languages, which are used to select four languages for training. The MAO-CHILDES corpus is selected based on the frequency of CDS, along additional considerations of evaluation.

lang	Samples	$\|V\|$	Tokens	Sentence Length μ	Children	Utterances
Chinese	857,792	518,172	850,510	258.28	949	3,293
German	582192	516,147	867,704	107.05	65	8105
Japanese	537,164	280,807	528,930	38.67	122	13,678
Indonesian	537,235	286,448	521,759	202.31	9	2,579
French	488,094	284,381	469,258	175.69	204	2,671
Spanish	332,903	211,559	331,009	167.85	291	1,972
Dutch	261,786	160,520	259,263	97.50	96	2,659
Portuguese	100,512	59,205	98,620	39.72	195	2,483
Polish	82,977	71,072	82,940	43.04	14	1,927
Swedish	80,936	53,719	79,739	49.34	6	1,616
Norwegian	55,262	31,310	40,215	32.62	6	1,233
Catalan	54,518	37,250	53,157	29.73	7	1,788
Romanian	33,130	20,700	32,986	16.58	6	1,990
Croatian	51,948	36,922	51,809	27.33	3	1,896
Czech	45,122	33,185	44,117	27.15	6	1,625
Danish	44,909	25,039	44,909	24.94	2	1,801
Bulgarian	31,715	21,435	31,714	32.76	1	968
Afrikaans	22,021	18,475	21,984	18.68	52	1,177
Irish	18,973	13,598	18,869	9.82	5	1,921
Russian	7,008	5,963	7,007	4.42	2	1,585
Icelandic	47,945	27,775	46,516	11.36	1	4,094
Slovenian	1,384	1,243	1,382	10.39	1	133
Thai	38,550	27,084	38,329	100.34	18	382

Figure 4: Distribution of Silver Tags across all languages in the MAO-CHILDES corpus, annotated using a SpaCy Multilingual UPOS Tagger

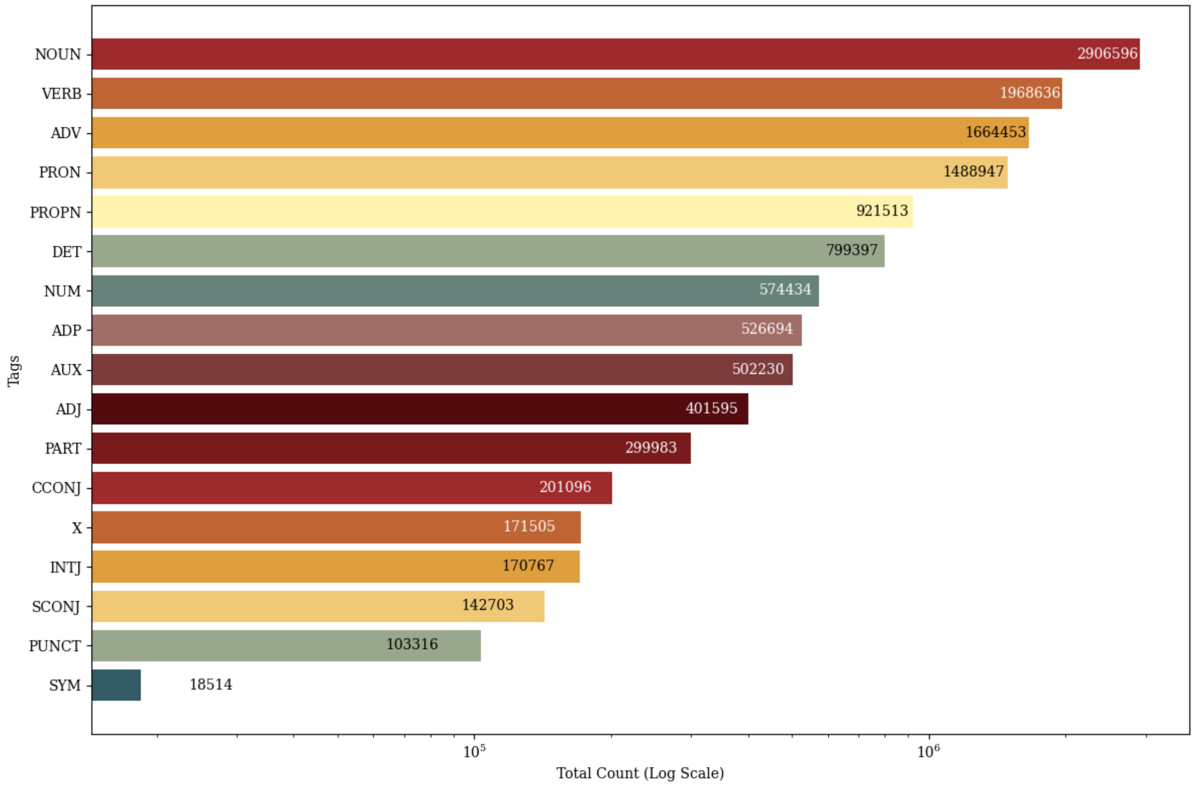


Table 7: **(English)** Evaluation of BabyBERTa model with four Cognitively-Plausible Curriculum Learning Strategies on BLIMP. English GROWING based on "Growing Trees" (Friedmann et al., 2021), INWARDS based on "Inward Maturation" (Heim and Wiltschko, 2021) and MMM (UPOS) and the language-specific *sem*-tag MMM (SEM) curricula based on Biberauer and Roberts (2015).

Grammatical Phenomenon	**Growing**	**Inwards**	**MMM** (UPOS)	**MMM** (SEM)
Anaphor	**96.22%**	84.67%	81.13%	**90.89%**
Arg Str	79.13%	79.86%	84.79%	**85.99%**
Binding	46.47%	71.75%	**83.42%**	77.76%
Control-Raising	77.03%	73.82%	**88.02%**	82.10%
Det-N Agreement	65.49%	65.19%	**84.38%**	79.31%
Ellipsis	58.24%	53.26%	42.77%	**70.94%**
Filler Gap	80.70%	**88.47%**	85.60%	73.11%
Irregular	**76.34%**	44.85%	54.42%	74.91%
Island	69.53%	62.87%	**96.62%**	68.64%
NPI	69.21%	76.02%	**83.42%**	74.13%
Quantifiers	44.54%	**84.79%**	58.43%	71.86%
Subject-Verb	65.98%	64.89%	**68.37%**	**79.03%**
Average Accuracy	71.13%	71.05%	74.22%	**77.35%**

Table 8: **(Chinese)** Comparison of accuracy of Chinese MAO-BABYBERTA ("vanilla") and GROWING, INWARDS, MMM (UPOS), MMM (SEM) objective curricula compared to a Chinese RoBERTa LLM baseline on the SLING minimal pairs dataset (Song et al., 2022)

Category	**Subcategory**	**Vanilla**	**LLM**	**Growing**	**Inwards**	**MMM (UPOS)**	**MMM (SEM)**
RelativeClause	rc_resumptive_pronoun	50.50	60.30	50.50	49.50	53.10	50.70
RelativeClause	rc_resumptive_noun	48.00	27.60	48.90	47.80	58.00	48.50
Anaphor	baseline_female	86.70	75.60	86.30	83.90	36.90	85.60
Anaphor	pp_female	70.50	71.80	70.80	67.50	41.80	69.80
Anaphor	baseline_male	12.50	38.50	45.20	45.30	81.90	45.20
Anaphor	Plural	51.98	97.95	53.10	51.20	52.33	52.10
Anaphor	self_male	14.30	92.60	47.80	46.10	81.40	46.90
Anaphor	pp_male	28.00	77.60	49.50	48.70	76.90	49.30
Anaphor	self_female	86.60	98.50	86.70	84.10	42.00	85.10
PolarityItem	any	54.20	85.60	55.30	52.70	49.20	54.60
PolarityItem	more_or_less	20.20	98.90	46.80	46.50	46.70	46.80
PolarityItem	even_wh	56.90	92.40	57.90	53.60	57.30	55.90
DefinitenessEffect	definiteness_every	85.70	94.60	85.40	83.30	88.50	84.20
DefinitenessEffect	definiteness_demonstrative	78.80	96.20	78.60	75.20	55.00	77.30
Aspect	zai_guo	49.30	97.30	49.70	47.90	43.10	49.20
Aspect	temporal_le	40.70	63.40	50.40	49.10	63.70	50.30
Aspect	zai_le	49.80	74.40	49.90	48.20	69.00	48.90
Aspect	temporal_guo	40.30	88.10	50.30	47.60	60.20	50.10
Aspect	zai_no_le	56.40	77.90	56.70	53.80	86.80	55.20
WhFronting	mod_wh	54.70	99.70	54.40	51.90	36.10	53.10
WhFronting	bare_wh	53.30	100.00	53.50	50.30	46.00	52.40
Classifier-Noun	cl_simple_noun	51.30	98.00	51.80	49.70	57.40	50.70
Classifier-Noun	cl_adj_simple_noun	52.60	96.30	52.10	50.10	61.80	51.30
Classifier-Noun	dem_cl_swap	51.10	99.60	51.20	49.20	60.70	50.60
Classifier-Noun	cl_adj_comp_noun	48.20	70.60	48.70	46.90	66.00	47.50
Classifier-Noun	cl_comp_noun_v2	49.60	88.80	49.30	47.30	61.90	48.80
Classifier-Noun	cl_comp_noun	51.00	72.00	51.60	49.80	61.30	50.90
Classifier-Noun	cl_adj_comp_noun_v2	52.20	89.50	52.50	50.70	60.90	52.10
AlternativeQuestion	haishi_ma	43.00	95.00	45.70	45.90	49.10	45.70
Average		51.32	83.41	56.23	54.27	58.79	55.48

Table 9: **(Japanese)** Accuracy of the "vanilla" SSLM for Japanese (MAO-BabyBERTa) trained on CDS and the best performing objective curricula +MMM on each phenomenon in the Japanese Benchmark of Linguistic Minimal Pairs (Someya and Oseki, 2023) compared to a Japanese monolingual GPT-2 LLM baseline trained on $\approx 30B$ words and a SSLM (WIKI) Baseline.

Phenomena	GPT2	WIKI	Vanilla	MMM
Control/Raising	16.67	50.00	25.00	70.00
Island Effects	75.76	64.00	72.06	92.19
Binding	58.97	79.05	57.86	89.62
NPI Licensing	50.00	83.33	75.00	90.00
Argument Structure	89.05	41.6	54.82	94.86
Ellipsis	85.96	49.36	56.13	97.68
Verbal Agreement	53.55	57.82	69.22	87.37
Filler-Gap	55.56	44.29	76.19	85.71
Morphology	82.86	49.77	55.08	82.05
Nominal Structure	95.65	41.51	55.87	92.12
Quantifiers	73.81	48.96	60.56	78.52
Average	**77.95**	**55.42**	**61.21**	**87.31**

Table 10: **(French and German CLAMS)** Performance of GROWING, INWARDS, MMM (UPOS) in French and MMM (UPOS) in German (the best performing objective curricula) on CLAMS (Mueller et al., 2020) compared to MAO-BABYBERTA SSLM ("vanilla") and the LLM and SSLM (WIKI) baselines. We report the LLM baselines obtained by Mueller et al. (2020) for mBERT in French and German, which does not report results for "within objective relative" (object rel within) as all focus verbs for that particular language and construction were out-of-vocabulary. Chance CLAMS accuracy is 0.5.

Language	Model	Average	S-V	Obj Rel (within)	Obj Rel (across)	VP Coord	Prep Animate	Subject Relative	Long VP Coord
FRENCH	LLM	83.00%	100.00	–	86.00	100.00	57.00	57.00	98.00
	WIKI	70.68%	67.48	73.40	73.80	71.27	66.80	70.80	71.27
	Vanilla	80.00%	82.0	64.90	84.8	78.6	84.8	83.1	82.1
	Growing	76.21%	73.70	69.57	79.51	71.12	86.53	80.01	73.70
	Inwards	79.01%	76.95	68.50	84.10	75.86	83.80	87.00	76.89
	MMM	75.93%	82.33	72.60	74.40	81.79	65.80	70.90	83.71
GERMAN	LLM	92.16%	95.00	–	93.00	97.00	95.00	73.00	100.00
	WIKI	59.63%	56.55	47.90	60.60	55.32	57.20	60.60	79.28
	MMM	73.25%	75.32	79.80	66.40	78.52	68.40	66.40	77.90

ConcreteGPT: A Baby GPT-2 Based on Lexical Concreteness and Curriculum Learning

Luca Capone[1*†] and **Alessandro Bondielli**[1,2†] and **Alessandro Lenci**[1†]

[1]CoLing Lab, Department of Philology, Literature and Linguistics, University of Pisa
[2]Department of Computer Science, University of Pisa

luca.capone@fileli.unipi.it, {alessandro.bondielli, alessandro.lenci}@unipi.it

Abstract

We present a model for the Strict-Small track of the BabyLM Challenge 2024 (Choshen et al., 2024). We introduce a Curriculum Learning approach for training a specialized version of GPT-2 (Radford et al., 2019), that we name ConcreteGPT. We utilize the norms from Brysbaert et al. (2014), which provide concreteness ratings for 40,000 English lexical items based on human subjects. Using these norms, we assign a concreteness score to each sentence in the training dataset and develop two curriculum strategies that progressively introduce more complex and abstract language patterns in the training data. Compared to the baselines, our best model shows lower performance on zero-shot tasks but demonstrates superior performance in fine-tuning tasks. Notably, our curriculum-trained models exhibit significant improvements over a non-curriculum based training of the same model.

1 Introduction

Optimising language model training to enhance efficiency without compromising performance presents a significant challenge, especially in the era of Large Language Models (LLMs) which require trillions of input tokens and millions of PetaFLOPs for training (Villalobos et al., 2024). A promising approach lies in exploring training strategies that streamline the learning process and maximise resource utilisation. Initiatives like the BabyLM Challenge (Warstadt et al., 2023) aim to find strategies to train effective LLMs under specific data constraints, that naturally reflect also on model sizes constraints following known scaling laws.

One possible area of interest in this context is the use of training strategies related to Curriculum Learning, which refers to the idea of training machine learning models on meaningfully ordered data, for instance from easier to harder samples (Bengio et al., 2009). This approach has yielded beneficial results on many tasks (Soviany et al., 2022), but has not been widely adopted in the context of language modelling. Typically, LLMs are trained on data scraped from the Web, for which it is difficult to obtain a meaningful ordering. In the present work, we evaluate the hypothesis that a curriculum learning strategy informed by evidence from human language acquisition can enhance model performance in data- and/or compute-constrained settings. Specifically, we attempt to understand the impact of considering **word concreteness** for ordering training data. In this context, concreteness refers to how tangible or perceptible the referent of a word is, with more concrete words being those that refer to physical objects or sensory experiences, while abstract words relate to concepts and ideas (Brysbaert et al., 2014). Word concreteness is often considered a proxy for the natural order in which children acquire language, beginning with words that represent familiar objects and situations (Bergelson and Swingley, 2013; Schwanenflugel, 2013). As language development progresses, children gradually learn terms that describe more complex concepts or relationships, which typically rely on the prior acquisition of simpler linguistic elements. Understanding the impact of word concreteness on language model training could potentially lead to models that better grasp and generate language in a more nuanced manner, and more importantly, that learn faster and more efficiently. While some studies have explored different language complexity metrics for Curriculum Learning (Opper et al., 2023; Mi, 2023; Martinez et al., 2023), none of the methods proposed in the 2023 BabyLM Challange employed curriculum

*Corresponding author

†For the specific purposes of Italian Academy, Luca Capone is responsible for Sections 2 and 5, Alessandro Bondielli is responsible for sections 3 and 4, Alessandro Lenci is responsible for sections 1 and 6

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 189–196
November 15-16, 2024 ©2024 Association for Computational Linguistics

criteria related to lexical concreteness (Warstadt et al., 2023).

In this work, we introduce a Curriculum Learning approach to train a specialised version of GPT2 (Radford et al., 2019), that we call **ConcreteGPT**, which leverages word concreteness ratings. We exploit the concreteness norms from Brysbaert et al. (2014), which include concreteness ratings obtained from human subjects for 40,000 lexical items in English. Using the norms, we compute a concreteness score for each sentence in the training dataset and create a curriculum that progressively emphasises more complex and abstract language patterns. We evaluate our approach on the Strict-Small track for the 2024 BabyLM Challenge. For the track, participants are provided with a dataset of 10M tokens for pre-training their model. Then, the model is evaluated in two ways: first, on a set of tasks in zero-shot settings, using Perplexity (PPL) or Pseudo-Log-Likelihood (PLL) metrics as a proxy of model understanding; second, the model is fine-tuned using standard fine-tuning or LoRA on the GLUE benchmark tasks. We evaluate two different models that employ a slightly different approach to building the curricula for the training, and compare them with a baseline model trained with the same amount of FLOPs without curriculum learning. We show that the curriculum-based models tend to outperform the non-curriculum model, while generally matching or slightly underperforming compared to the strong baselines provided by the task organizers (i.e., the winning models from the 2023 edition), despite a possibly lower computational cost.

The paper is organised as follows. First, we outline the motivations behind our curriculum learning approach in Section 2. Section 3 details the methodology used to create the datasets and describe the curriculum design. Sections 4 and 5 provide an in-depth discussion of the model, covering training specifics and results, and discuss the impact of two variations of the curriculum learning strategy. Finally, Section 6 draws some conclusions and highlights possible future directions.

2 Motivation Behind Curriculum Design

The motivation behind this approach stems from the hypothesis that a curriculum guided by word concreteness can enhance the model's learning trajectory by starting with more concrete, easily grasped examples and gradually advancing to more sophisticated verbal items. This method aims to improve the model's ability to handle a broader range of linguistic phenomena, potentially leading to more robust and contextually aware text generation. Given that the model is not multimodal, one might initially question the value of using a Curriculum Learning approach based on lexical concreteness, since the representations learned by the model are not grounded in perceptual experiences. It is useful to start from the assumption that, in principle, all meanings can be considered as abstract, referring to general classes capable of subsuming heterogeneous and always particular phenomena (Eco, 1979, §2.6). From this perspective, the value of an approach based on lexical concreteness does not lie in grounding meanings in perceptual experience (Søgaard, 2023). Despite the abstract character of meaning, it is widely accepted that the first words children learn tend to have a tight connection to their experience with referents (Schwanenflugel, 2013; Bergelson and Swingley, 2013). Early language acquisition typically involves words related to the child's surroundings, such as parents, pets, and daily routines objects. From these familiar meanings, children gradually expand their vocabulary to include words with more complex meanings that concern more abstract situations, require greater linguistic competence and larger cultural experience. Thus, concreteness rating can be understood as an index of the difficulty in acquiring a word. For instance, learning a term like "dog" requires less linguistic knowledge and semantic structuring than understanding a more abstract concept like "justice". By initially exposing the model to sentences containing words with an high concreteness rating, we attempt to simulate this learning trajectory, providing the model with simpler, more fundamental contexts before progressing to the acquisition of more complex meanings and linguistic situations. The method proposed in this paper is consistent with the findings of Abdou et al. (2021) and Patel and Pavlick (2022) which suggest that LM embeddings encode perceptual structures (e.g, meaningful spatial relations and colors) without requiring perceptual grounding. The hypothesis is that a curriculum based on lexical concreteness can facilitate the acquisition of these meaningful structures.

A potential objection to our method is why age of acquisition is not used directly as a feature for sorting the curriculum. The core principle of the Curriculum Learning approach is to establish a criterion that accurately assesses the difficulty of language items, thereby grouping items of similar difficulty together. In this context, a criterion such as age of acquisition alone does not serve this purpose effectively. Linguistically similar items — those that function similarly in speech and possess comparable levels of difficulty — can exhibit different ages of acquisition. In fact, consulting data from WordBank (Frank et al., 2017), specifically the British Oxford Communicative Development Inventory (CDI), we observe that linguistically similar items are acquired at different ages by children. For example, the proportion of children understanding the word *dog* at 12 months is slightly over 0.6, whereas the corresponding value for the word *lamb* is just under 0.1. This disparity remains relatively constant until 25 months, even though *lamb* does not appear to present any particular challenges compared to *dog*. Consequently, two words of similar difficulty may be sorted differently within the curriculum based solely on age of acquisition. In contrast, this issue is mitigated by the use of concreteness ratings, with *lamb* being rated at 4.97 and *dog* at 4.85 in Brysbaert et al. (2014).

3 Curriculum Design

Brysbaert et al. (2014) collected ratings from 4,237 native speakers for 37,058 English words and 2,896 two-word expressions. The ratings ranged from a minimum of 1, representing «something you cannot experience directly through your senses or actions», to a maximum of 5, indicating «something that exists in reality; you can have immediate experience of it through your senses (smelling, tasting, touching, hearing, seeing) and the actions you do. The easiest way to explain a word is by pointing to it or by demonstrating it» (Brysbaert et al., 2014). Based on these ratings, we assigned a concreteness score to each sentence in the dataset (10M Strict-Small dataset, Choshen et al. 2024). For each sentence, only adjectives, nouns, and verbs were considered in the score calculation. The concreteness ratings of the words in the sentence were summed, and the total was divided by the number of selected words. This resulting value corresponds to the **sentence concreteness score**. Once the sentence scores were obtained, the dataset was divided into four slices, each containing approximately 300,000 items, based on increasing concreteness, as shown in Figure 1.

Based on the dataset slicing, we devise two different curriculum strategies:

SEQUENTIAL – This strategy considers the slicing as-is, and the curriculum is based on their sequential ordering, from the most concrete to the most abstract.

MIXED – This strategy is more nuanced, and accounts for the fact that while sentence-level concreteness can be used as proxy for the natural order in which children acquire language, it is also likely that childrens will be exposed to more complex words as well. Thus, starting from the original slices, we redistribute part of each slice into the other ones. Specifically, each slice contains 50% of the data from the original slice, and 50% from the other three slices, in different proportions, to simulate an increasing percentage of progressively more abstract sentences in each slice. The exact proportions of sentences from each mixed slice are reported in Figure 2.

4 Model and Training

In our experiments, we use the GPT2 implementation from HuggingFace[1] as our base architecture, with its standard pretrained tokenizer. The model has 124M trainable parameters. To further limit the computational cost of training, we restrict the context length of the model to 128 tokens. This change is driven not only by concerns regarding computational resources, but also by theoretical considerations related to the development of working memory in humans, which appears to be limited during the early years of life (Swanson, 1996; Cowan, 2016). A reduced context length (though still larger than the number of tokens a child can process) better aligns with the cognitive plausibility criteria required by the challenge.

We experiment with a hybrid training procedure where the model is sequentially trained on each slice of the dataset for three epochs with the same hyperparameters. Note that we restart the training procedure each time, and that we randomized the batch sampling within the training slice. This means that data from each batch is randomly sampled (as customary for training LMs) from the training slice. Then, the resulting model is further trained on the entire dataset with a lower learning rate for an additional two epochs, again with random batch sampling. We follow this procedure for

[1] https://huggingface.co/openai-community/gpt2

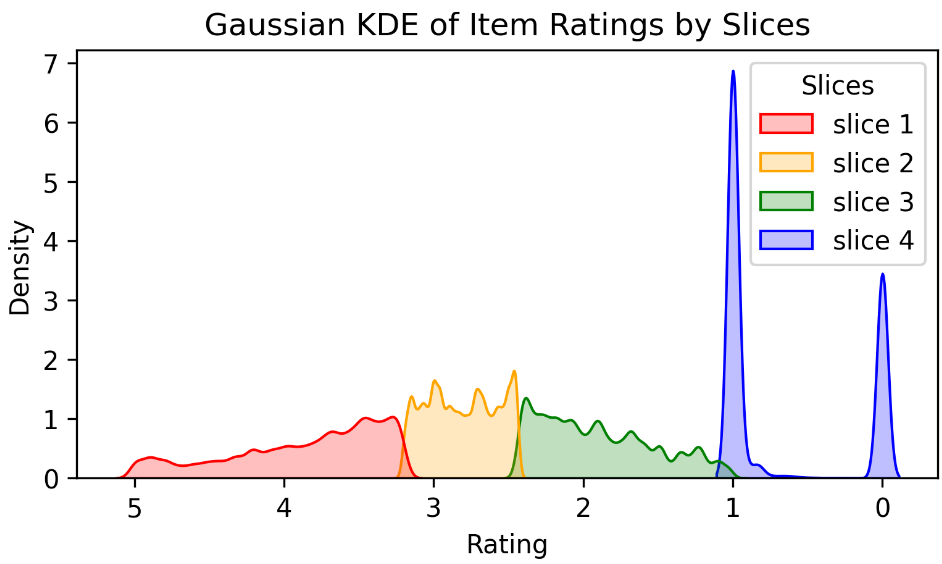


Figure 1: Distribution of sentences into slices. The dataset contains sentences with adjectives, verbs, and nouns that are not among the rated words (assigned a score of zero), as well as sentences that contain none of these word types (assigned a score of one). All such sentences are grouped into the final slice, representing the most abstract and complex sentences.

Model	Data	Epochs	Init. LR	LR scheduler	Batch size	Grad. accum.	Warmup
SEQUENTIAL	Slice 1	3	5e-4	Cosine	32	8	1000
	Slice 2	3	5e-4	Cosine	32	8	1000
	Slice 3	3	5e-4	Cosine	32	8	1000
	Slice 4	3	5e-4	Cosine	32	8	1000
	Full Dataset	2	2e-4	Cosine	32	8	1000
MIXED	Slice 1 - mix	3	5e-4	Cosine	32	8	1000
	Slice 2 - mix	3	5e-4	Cosine	32	8	1000
	Slice 3 - mix	3	5e-4	Cosine	32	8	1000
	Slice 4 - mix	3	5e-4	Cosine	32	8	1000
	Full Dataset	2	2e-4	Cosine	32	8	1000
SHUFFLE	Full Dataset	5	5e-4	Cosine	32	8	1000

Table 1: Pre-training parameters for each of the models. In the case of curriculum-based models, parameters are reported for each slice.

Hyperparameter	Value
Initial learning rate	3e-4
Batch size	64
Maximum epochs	32
Evaluate every (epochs)	1
LoRA alpha	16
LoRA rank	8
LoRA dropout	0.1

Table 2: Parameters for fine-tuning with LoRA on the GLUE tasks.

both the SEQUENTIAL and MIXED models, only changing the composition of the slices as described in Section 3. For training the comparison model (i.e., the model without curriculum learning), that we call SHUFFLE, we aimed to use the same amount of computing, and thus to show the model each data point the same number of times. Therefore, we trained it for 5 epochs on the entire dataset with random sampling.

Table 1 summarizes the training parameters. All models were trained using half precision (fp16). No direct hyperparameter optimization was performed. However, we experimented with several configurations, specifically varying the initial Learning Rate and its scheduler, and found the chosen configuration to work best. As for batch size and gradient accumulation steps, the values were chosen to best fit the available computational resources. All models were trained using a Nvidia

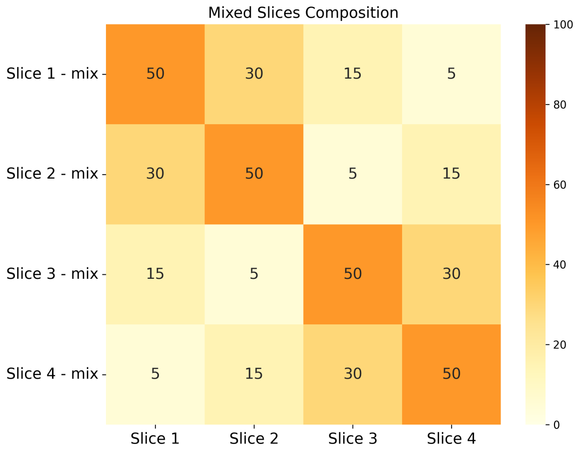


Figure 2: Percentage of sentences from each slice for training the MIXED model.

A100 40GB GPU. Notably, we used the same random seed for all training runs, to ensure that the starting condition was the exact same for all of the trained models. We are aware that averaging the results of multiple training runs would have yielded more reliable results. However, it would have also drastically increased the computational cost of our experiments. The pre-training procedure was handled with the HuggingFace Trainer.[2] For the fine-tuning, we train a LoRA for each of the GLUE tasks using the script provided by the challenge organisers. As for the hyperparameters, we left the default ones provided by the challenge organisers (Choshen et al., 2024). For the sake of completeness, we report the LoRA fine-tuning parameters in Table 2.

5 Results and Discussion

The models are evaluated on two distinct sets of tasks: one requiring fine-tuning and the other performed in a zero-shot setting. Fine-tuning was conducted using the script provided by the organisers (see Section 4). The baselines are two models trained by the organizers, and inspired by the 2023 edition winning systems: LTG-BERT (Charpentier and Samuel, 2023) and BabyLlama (Timiryasov and Tastet, 2023).

For the fine-tuning task, models are fine-tuned on the GLUE benchmark tasks (Wang et al., 2018). Table 3 shows results on all the tasks for each of our trained models, namely SEQUENTIAL, MIXED, and SHUFFLE. While the differences in performance across the models are not substantial, the curriculum-based models (SEQUENTIAL and MIXED) consistently outperform the non-curriculum one (SHUFFLE), with the exception of the CoLA and RTE tasks. Among curriculum-based models, the MIXED model outperform the SEQUENTIAL model on 7 out of 10 tasks, and for 2 out of 10 taks they achieve the same level of performances.

For the zero-shot tasks, results are reported in Table 4, and are less clear-cut. For the Ewok task (Ivanova et al., 2024), the MIXED model perform slightly better than SHUFFLE and SEQUENTIAL models, and achieve a score on par with the best baseline. For the Blimp tasks (Warstadt et al., 2020) the scenario is different: in Blimp Filtered, none of our models manage to match the BabyLlama baseline, although the two models trained with curriculum learning come very close. Nevertheless, they significantly outperform LTG-BERT. For Blimp Supplement, none of the models reach the baseline, and the best-performing model is the non-curriculum one (SHUFFLE). Nevertheless, in two out of three zero-shot tasks curriculum-based models outperform, albeit slightly, the non-curriculum based one. In Blimp Filtered, both models perform the same, while for Ewok the best performing model is again the MIXED model. Table 4 also report the average on the fine-tuning GLUE tasks. The MIXED model significantly outperform both the other proposed models as well as the baselines.

On average, across all tasks, the three models consistently outperform the worse baseline (LTG-BERT), but do not exceed the performance of the best baseline (BabyLlama) except for GLUE. Of the three, the MIXED model performs the closest to BabyLlama overall. However, this result is largely influenced by the poorer performance in Blimp, especially in Blimp Supplement. In this case the curriculum learning strategy appears to negatively affect performance on acceptability tasks, as indicated by the weaker results observed in CoLA and Blimp. A potential explanation is that curriculum learning based on lexical concreteness enhances performance in tasks with a stronger semantic component, such as MRPC, SST-2, and MNLI, where the curriculum-trained models demonstrate superior performances.

However, these findings appear to corroborate the idea that, given the same (and limited) amount of data and training compute, employing a cognitively plausible training strategy that leverages lexical concreteness as a proxy for a plausible or-

[2] https://huggingface.co/docs/transformers/v4.44.2/en/main_classes/trainer.

model	mrpc	boolq	qqp	sst2	qnli	wsc	cola[matt_corr]	rte	mnli	multirc
SEQUENTIAL	0.80	0.64	**0.77**	0.80	0.74	0.58	0.59 [0.02]	**0.57**	0.67	**0.65**
MIXED	**0.82**	**0.66**	**0.77**	**0.85**	**0.78**	**0.65**	0.61 [0.04]	0.56	**0.68**	**0.65**
SHUFFLE	0.79	**0.66**	0.74	0.79	0.76	0.62	0.59 [**0.08**]	**0.57**	0.64	0.64

Table 3: Fine-tuning results. As specified in the evaluation pipeline documentation (github.com/babylm/evaluation-pipeline-2024), we use accuracy as the evaluation metric for all tasks except QQP and MRPC, for which we report F1 scores, and CoLA, for which we use the Matthews correlation coefficient (we also report the evaluation loss for this task).

Model	Zero Shot			Fine tuning	Macro Avg.
	blimp_supp	blimp_filt	ewok	Glue Avg.	
SEQUENTIAL	55.9	**68.6**	50.2	62.4	59.3
MIXED	55.9	**68.6**	**50.7**	**64.6**	**60.0**
SHUFFLE	**57.1**	67.8	50.5	62.9	59.6
BabyLlama	59.5	**69.8**	**50.7**	63.3	**60.8**
LTG-BERT	**60.8**	60.6	48.9	60.3	57.7

Table 4: Overall results and comparison with baselines.

dering to acquire language is probably beneficial. In addition to this, it is also relevant to point out that our model, albeit larger in terms of number of parameters, was not trained until convergence and in any case was trained with less computing than the strongest baseline represented by the BabyLlaMA model, but it still either reach its performances or surpasses them in 2 out of the 4 evaluations.

6 Conclusion and Future Work

In this paper we propose two models for the Strict-Small track of the BabyLM Challenge 2024 (Choshen et al., 2024). The models were trained using a Curriculum Learning strategy designed to optimise performance. The dataset provided by the organisers was divided into four slices based on increasing levels of lexical concreteness. From this division, two models were trained: the SEQUENTIAL was trained on the slices in order of decreasing concreteness, while the MIXED incorporated a progressively higher percentage of abstract and complex sentences at each epoch. For comparison, the same architecture was trained using a standard training procedure on the entire dataset with the same amount of compute (SHUFFLE model).

The SHUFFLE model outperforms the curriculum-trained models only in the Blimp Supplement (for zero-shot) and in CoLA (for fine-tuned) tasks. In all other tasks however curriculum learning based on lexical concreteness, particularly the MIXED model, demonstrates improved performance. Compared to the baselines provided by the organisers, the MIXED model exhibits comparable or lower performance on zero-shot tasks but performs well in fine-tuning tasks. These results are notable, especially given the relatively small amount of training compute provided to the model.

Our findings suggest that in low resources and/or low compute scenarios, cognitively plausible training strategies, specifically using concreteness, may help the model learn effective representation faster than with traditional training methods. Nevertheless, we must point out that the proposed approach does not systematically outperform the strong baselines provided by the challenge organisers, especially in zero-shot tasks. Possible explanations are that i.) our concreteness-based approach still requires some refinement, and that ii.) our models may be undertrained with respect to the baselines.

Based on these findings, we propose several directions for future work. First, training the model on a larger dataset and for more epochs would allow us to test whether the performance gap scales with additional data, potentially by further refining the progression of the slices in the MIXED strategy. Second, applying this curriculum learning approach to a multimodal model would help assess whether it also facilitates mapping between language and images. Finally, it would be valuable to further investigate the differences in performance on acceptability tasks (which are more syntactic in nature) versus tasks focused on semantics and inference, to better understand the robustness of this trend.

Acknowledgments

We acknowledge financial support under the PRIN 2022 Project Title "Computational and linguistic benchmarks for the study of verb argument structure" – CUP I53D23004050006 - Grant Assignment Decree No. 1016 adopted on 07/07/2023 by the Italian Ministry of University and Research (MUR). This work was also supported under the PNRR—M4C2—Investimento 1.3, Partenariato Esteso PE00000013—"FAIR—Future Artificial Intelligence Research"—Spoke 1 "Human-centered AI," funded by the European Commission under the NextGeneration EU programme, and partially supported by the Italian Ministry of University and Research (MUR) in the framework of the PON 2014-2021 "Research and Innovation" resources – Innovation Action - DM MUR 1062/2021 - Title of the Research: "Modelli semantici multimodali per l'industria 4.0 e le digital humanities."

References

Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. 2021. Can language models encode perceptual structure without grounding? a case study in color. *arXiv preprint arXiv:2109.06129*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning (icml). *Google Scholar Google Scholar Digital Library Digital Library*.

Elika Bergelson and Daniel Swingley. 2013. The acquisition of abstract words by young infants. *Cognition*, 127(3):391–397.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46:904–911.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts bert. *arXiv preprint arXiv:2311.02265*.

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*.

Nelson Cowan. 2016. Working memory maturation: Can we get at the essence of cognitive growth? *Perspectives on Psychological Science*, 11(2):239–264.

Umberto Eco. 1979. *A theory of semiotics*, volume 217. Indiana University Press.

Michael C Frank, Mika Braginsky, Daniel Yurovsky, and Virginia A Marchman. 2017. Wordbank: An open repository for developmental vocabulary data. *Journal of child language*, 44(3):677–694.

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

Richard Diehl Martinez, Zebulon Goriely, Hope McGovern, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. Climb: Curriculum learning for infant-inspired model building. *arXiv preprint arXiv:2311.08886*.

Maggie Mi. 2023. Mmi01 at the babylm challenge: Linguistically motivated curriculum learning for pretraining in low-resource settings. *Proceedings of the BabyLM Challenge. Association for Computational Linguistics (ACL)*.

Mattia Opper, J Morrison, and N Siddharth. 2023. On the effect of curriculum learning with developmental data for grammar acquisition. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 346–355.

Roma Patel and Ellie Pavlick. 2022. Mapping language models to grounded conceptual spaces. In *International conference on learning representations*.

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.

Paula J Schwanenflugel. 2013. Why are abstract concepts hard to understand? In *The psychology of word meanings*, pages 223–250. Psychology Press.

Anders Søgaard. 2023. Grounding the vector space of an octopus: Word meaning from raw text. *Minds and Machines*, 33(1):33–54.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *Int. J. Comput. Vision*, 130(6):1526–1565.

H Lee Swanson. 1996. Individual and age-related differences in children's working memory. *Memory & Cognition*, 24(1):70–82.

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*.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. 2024. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, et al. 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*.

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.

When Babies Teach Babies: Can student knowledge sharing outperform Teacher-Guided Distillation on small datasets?

Srikrishna Iyer
Artificial Intelligence - Data Analytics Strategic Technology Center,
ST Engineering IHQ Ltd., Singapore
srikrishna.rameshiyer@stengg.com

Abstract

We present our submission[1][2] to the BabyLM challenge, aiming to push the boundaries of data-efficient language model pretraining. Our method builds upon deep mutual learning, introducing a student model search for diverse initialization. We address the limitation of treating students equally by formulating weighted mutual learning as a bi-level optimization problem. The inner loop learns compact students through online distillation, while the outer loop optimizes weights for better knowledge distillation from diverse students. This dynamic weighting strategy eliminates the need for a teacher model, reducing computational requirements. Our evaluations show that teacher-less methods can match or surpass teacher-supervised approaches.

1 Introduction

The substantial computational and memory requirements of large language models pose significant challenges for deployment on intelligent edge systems, where resources are often constrained. As the demand for real-time processing and low-latency responses increases in edge computing environments, the need for lightweight and memory-efficient models becomes critical. Recent research, notably the Chinchilla paper (Hoffmann et al. (2024)), demonstrated that a 70B parameter model trained on 1.4 trillion tokens outperformed larger models with less data, highlighting the intricate balance between model size and training data. This massive data requirement—equivalent to over 10,000 times the words a 13-year-old encounters—is becoming a significant bottleneck. To address these challenges, several techniques have emerged such as network pruning (Han et al. (2015)), quantization (Courbariaux et al. (2015)), neural architecture search Ren et al. (2021) and Knowledge distillation (Hinton et al. (2015),Li et al. (2020),Wang et al. (2022))

In response to these challenges, the BabyLM challenge invites researchers to explore the limits of data-efficient language model pretraining (Choshen et al.). Participants are constrained to training their models on limited text corpora: 10M and 100M word text-only tracks and a newly introduced multimodal track containing 50M words of paired text-image data, and 50M words text-only data.

Our paper describes our submission to the 10M and 100M text-only tracks. It builds upon the approach of weighted mutual learning Zhang et al. while introducing key modifications to enhance generalizability. Our methodology focuses on distilling a RoBERTa-base model (125M parameters) to less than half its size while maintaining performance. Our main contributions include :

- We use Bayesian optimization to select model architectures of student models by varying hidden layers, attention heads, and hidden sizes.
- Instead of the traditional teacher-student distillation, we explore weighted mutual learning through a bi-level optimization process : (a) The inner loop minimizes a combined loss to train individual student models, consisting of a supervised learning loss and a KL divergence loss that aligns each student's class posterior with others'. (b) Instead of treating each student model equally, we introduce an outer loop to optimize student importance weights by minimizing the ensemble loss.

This approach generally performed better than both conventional supervised learning and traditional distillation from a larger pretrained teacher. Notably, our weighted mutual learning strategy can

[1] https://huggingface.co/AI-DA-STC/RoBERTa_WML_distill-Babylm-10M-2024
[2] https://github.com/AI-DA-STC/generative-ai-research-babylm

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 197–211
November 15-16, 2024

improve performance even among several large networks compared to independent learning, challenging the conventional understanding that distillation requires a larger, more powerful teacher.

2 Related Work

The vanilla distillation Hinton et al. (2015) method consists of two stages, firstly train a large teacher model, followed by transfer of soft logits to a smaller student model. Also known as Offline distillation, it keeps the teacher fixed, only allowing a one-way knowledge transfer. To reduce memory consumption of training a large teacher model, Zhang et al. (2018) proposed an online distillation framework called mutual learning where a group of student (or student) models were trained simultaneously. Although, online distillation eliminated the teacher model, similar networks in online distillation may prevent the students from learning knowledge from the students Zhang et al.. Recent approaches have attempted to induce diversity in online distillation to improve overall performance. Chen et al. (2020) proposed inducing data diversity by training student models with varying image augmentations. However, this method relies heavily on data augmentations, which can be unpredictable in real-world deployment scenarios. Du et al. (2020) introduced an adaptive ensemble knowledge distillation method using multiple diverse teacher models to train a student model. While this approach shows promise, it requires maintaining several teacher models, leading to increased memory usage and computational overhead. The reported accuracy improvements are also relatively modest, typically ranging from 0.5% to 1% across benchmarks. Our approach closely resembles to that of Zhang et al.. They present a diversity induced weight mutual learning approach for distillation. They introduce diversity by assigning varying pruning ratios to different student models. Although this method reduces memory consumption, the manual assignment of pruning ratios may not generalize well across different architectures and tasks. The reported performance gains are limited, with improvements of less than 0.5% on most benchmarks. As shown by Liu et al. (2017), while pruning induces sparsity within networks and can reduce computational complexity (measured in FLOPs), the relationship between pruning percentage and actual model size reduction is not always linear. Moreover, in Zhang et al., we observe a performance drop when pruning beyond 30%, indicating a trade-off between model compression and accuracy.

3 Diversity Induced Weighted Mutual Learning

3.1 Diversifying student models

In our approach to create diverse student models for the Diversity Induced Weight Mutual Learning (DWML) framework, we employ Bayesian optimization to efficiently search for optimal architectural configurations. Given a teacher model with N parameters, we aim to generate p student models, where the i-th student model targets approximately N_i parameters, defined as:

$$N_i = \frac{N}{i+1}, \quad i \in 1, 2, ..., p \tag{1}$$

This optimization problem can be formally defined as finding, for each student i, an architecture a_i from the set of all possible RoBERTa architectures A that minimizes $||params(a_i) - N_i||$, where $params(a_i)$ represents the parameter count of architecture a_i. We chose Bayesian optimization for this task due to its efficiency in exploring high-dimensional spaces with relatively few function evaluations, making it less computationally expensive compared to alternative methods such as grid search or random search (Kandasamy et al., 2018). Our implementation utilizes the BayesianOptimization library (Nogueira, 2014–), with a search space encompassing the number of layers, number of attention heads, and embedding dimension. The objective function calculates the difference between the actual parameter count of a given architecture and the target parameter count, with a constraint ensuring the embedding dimension is divisible by the number of attention heads.

3.2 Weighted Mutual Learning using Bi-level optimisation

Building upon the work of (Zhang et al.), we introduce a modified approach to Weighted Mutual Learning using bi-level optimization. Our method replaces the pruning-based initialization with Bayesian optimization for student model selection.

The overall loss function for training M peer

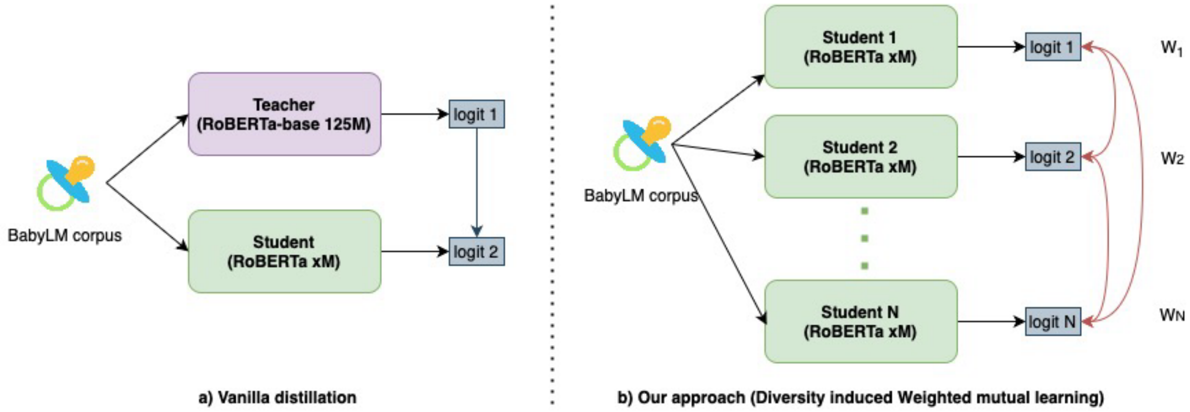


Figure 1: Overview of the difference between Vanilla knowledge distillation and our approach, Diversity induced weighted mutual learning (DWML). (a) Hinton et al. (2015) is the popular knowledge distillation method, where the student network (RoBERTa-xM) can only learn from a trained teacher network (RoBERTa-base-125M). Here xM refers to a student model of x million parameters. (b) is the Diversity Induced Weight Mutual Learning (DWML) framework where each student model is initialised with parameter counts = $N/2, N/3..N/(p+1)$ using Bayesian optimisation search. Rather than averaging the knowledge from students, DWML leverages bi-level optimization to estimate the relative importance of each student (e.g., weight ω_i for student i).

models is defined as:

$$\begin{aligned} \text{loss} &= (1-\alpha)\sum_{i=1}^{M}\omega_i L_{CE}(z_i, Y) \\ &+\alpha\sum_{i=1}^{M}\sum_{j=1}^{M}\omega_j KL(z_i, z_j) \end{aligned} \tag{2}$$

where ω_i indicates the importance of the i-th student model, α balances the supervision from labels and peers, L_{CE} is the cross-entropy loss, and KL is the Kullback-Leibler divergence. ω_j is the importance of every other student model except the i-th one. Both z_i and z_j are model logits. We formulate the weighted mutual learning as a bi-level optimization problem. The inner loop optimizes the network parameters θ using the loss in equation 2. As shown in the paper, the gradient for the outer loop optimization, also known as the hypergradient, is calculated as:

$$g_{\omega_i} = \nabla_{\omega_i} L_2 = \frac{\partial L_2}{\partial \omega_i} - \gamma \frac{\partial L_2}{\partial \theta}\frac{\partial L_a}{\partial \theta}^T \tag{3}$$

where $L_a = (1-\alpha)L_{CE}(z_i, Y) + \alpha\sum_{j=1}^{M} KL(z_j, z_i)$ is the ensemble loss. Since ω is a probability simplex that $\sum_{i=1}^{M}\omega_i = 1$, we use the mirror descent to update ω [3, 5]. Algorithm 1 outlines our weighted mutual learning for online distillation. To be more specific, we first run several steps of gradient descent based on the loss function in 2 to update model parameters θ with a fixed ω. Then we calculate the gradient of ω_i based on 3, and run one step of mirror descent to update ω_i:

$$\omega_i^{k+1} = \frac{\omega_i^k \exp\{-\eta\nabla_{\omega_i^{k+1}} L_2\}}{\sum_{i=1}^{M}\omega_i^k \exp\{-\eta\nabla_{\omega_i^{k+1}} L_2\}} \tag{4}$$

where η is the step size with annealing, and ω_i^k is the importance of the i-th peer in the k-th step.

4 Training

4.1 RoBERTa-base

Our models are based on RoBERTa-base (Liu et al., 2019). This model has shown reasonably good performance on small text corpus. We use the raw RoBERTa-base as a baseline in the evaluations. We use it as our teacher model to distill student models using knowledge distillation (KD) and the teacher supervised version of weighted deep mutual learning (KD_DWML). Details about the hyperparameters found from the search are shown in 3. The models were pre-trained (and finetuned for GLUE, SuperGLUE tasks) using 1 Nvidia H100 GPU with 80GB VRAM.

4.2 Dataset

We pretrain all our language models on the 10M and 100M datasets of the BabyLM challenge from 2023 (Warstadt et al., 2023). We adopt the same preprocessing pipeline from (Samuel et al., 2023)

Algorithm 1: Diversity Induced Weighted Mutual Learning (DWML)

Input: Dataset $\{(x_n, y_n)\}_n^N$; Number of peers M; Teacher model size N

1: Define parameter space Θ for number of layers,attention heads, and hidden size
2: Objective function $f(\theta) = |\text{params}(\theta) - N_i|$ where $N_i = N/(i+1)$
3: **for** $i = 1$ to M **do**
4: Use Bayesian optimization to find optimal θ_i^* from Θ
5: Initialize peer model i with parameters θ_i^*
6: **end for**
7: Initialize peer weights ω^0
8: **for** $k = 1$ to K **do**
9: With peer importance ω^k, run T steps of AdamW to update model parameters θ using Eq. 2
10: Calculate gradient for ω^k based on Eq. 3
11: Update ω^k to ω^{k+1} using mirror descent with Eq. 4
12: **end for**

Output: M models with outputs $z_1, ..., z_M$ and weights for peers ω

for standardizing the text corpus. The detailed breakdown of the datasets are shown in 5. The reason why we select the datasets from 2023 is that it appears to be similar to the dataset released for the 2024 challenge (Choshen et al.). The only difference is the exclusion of the QCRI Educational Domain (QED) Corpus and higher proportion of CHILDES from 4.21M to 29M. This was done because the QED was of poor quality. However, we believe that the 2023 dataset gives us an opportunity to explore how distilled models perform when trained datasets that closely represent real world textual data that is unavoidably noisy.

5 Results

This section provides the results of the empirical evaluation of DWML. First, we compare our method to baselines, then we compare our method with other distillation methods and then we perform an ablation study of different DWML variations.

5.1 BabyLM Challenge evaluation

We use the BabyLM evaluation pipeline to assess our models. This pipeline measures syntactic understanding through the Benchmark of Linguistic Minimal Pairs (BLiMP & BLiMP supplemental,Warstadt et al. (2020)). It evaluates general knowledge using the Elements of World Knowledge (EWoK, Ivanova et al. (2024)) benchmark. For overall natural language understanding, it uses GLUE (Wang et al. (2018)) and SuperGLUE (Wang et al. (2019). If applicable, we divide the training set into a train-development split and report the mean statistics over multiple runs on the hidden validation split.The detailed scores are shown in section D

Text-only 10M Dataset				
Model	BLiMP	Supp.	EWoK	GLUE
BabyLlama	**69.8**	59.5	50.7	**63.3**
LTG-BERT	60.6	**60.8**	48.9	60.3
RoBERTa-base	49.6	48.9	**51.6**	42.5
RoBERTa-DWML	51.6	52.3	50.3	43.1
Text-only 100M Dataset				
Model	BLiMP	Supp.	EWoK	GLUE
BabyLlama	**73.1**	60.6	52.1	**69.0**
LTG-BERT	69.2	**66.5**	51.9	68.4
RoBERTa-base	49.8	46.8	50.25	43.4
RoBERTa-DWML	52.1	48.4	51.6	44.0

Table 1: Results for the BabyLM challenge evaluation datasets. We compare our submitted model (RoBERTa-DWML) to the base model (RoBERTa-base) and the baselines given by the organizers of the challenge on the 10M and 100M datasets.

BLiMP Our RoBERTa-DWML demonstrates consistent improvements over RoBERTa-base across both dataset sizes. On the 10M dataset, DWML achieves 51.6% compared to RoBERTa-base's 49.6%, showing a 2% improvement. This gain is maintained in the 100M dataset, where DWML scores 52.1% versus RoBERTa-base's 49.8%. While these improvements are modest, they demonstrate that our teacher-less approach can enhance syntactic understanding with minimal computational overhead. It's worth noting that BabyLlama's multi-teacher distillation approach (Timiryasov and Tastet, 2023) significantly outperforms all models (73.1% on 100M), though this comes at the cost of substantial computational requirements in maintaining and training with multiple teacher models (GPT-2 and LLaMA), which may not be practical for resource-constrained applications.

BLiMP Supplemental The supplemental BLiMP results further validate the effectiveness of our DWML approach. For the 10M dataset, RoBERTa-DWML (52.3%) outperforms

RoBERTa-base (48.9%) by a margin of 3.4%. In the 100M setting, we observe a similar trend with DWML (48.4%) showing improvement over the base model (46.8%). These consistent gains come with minimal additional computational cost over the base model. While BabyLlama achieves substantially higher performance (60.6% on 100M), this improvement requires significant computational resources for managing multiple teacher models during training and inference, a trade-off not examined in their original work.

EWoK On the world knowledge tasks, RoBERTa-DWML maintains competitive performance relative to RoBERTa-base. In the 10M dataset, DWML (50.3%) performs slightly below the base model (51.6%), while in the 100M dataset, DWML (51.6%) shows improvement over the base model (50.25%). These results demonstrate the capability of our lightweight approach in preserving world knowledge. While BabyLlama leads with 52.1% on the 100M dataset through its multi-teacher architecture, the relatively small performance gap (0.5%) raises questions about whether the significant computational overhead of maintaining multiple teacher models is justified for world knowledge tasks in resource-constrained environments.

GLUE All the models were fine-tuned on the GLUE and SuperGLUE datasets and then evaluated on their linguistic performance. On the GLUE benchmark, RoBERTa-DWML shows marginal improvements over RoBERTa-base across both dataset sizes. For the 10M dataset, DWML achieves 43.1% compared to RoBERTa-base's 42.5%, representing a modest 0.6% gain. This pattern continues in the 100M setting, where DWML (44.0%) slightly outperforms the base model (43.4%). These results suggest that our teacher-less approach maintains general language understanding capabilities

5.2 Comparison with Other Distillation Methods

To evaluate the effectiveness of our proposed distillation method, in Table 2 we compare its performance against other distillation techniques using accuracy scores. Our framework is compared to Self-Distillation (SD, Zhang et al. (2019)), a method that allows a small-sized student model to distill knowledge within its network. Knowledge distillation (KD,Hinton et al. (2015)) is the vanilla distillation framework that uses a student network to approximate the output logits of a pre-trained teacher network. Deep mutual learning (DML,Zhang et al. (2018)) an ensemble of students learn collaboratively (without a teacher) and teach each other. The main difference between DML and our diversity induced weight mutual learning (DWML) framework is the usage of dynamically learned student weights using a bi-level optimization objective.Knowledge distillation based diversity induced weight mutual learning (KD_DWML) is the teacher-supervised version of DWML. The GPU utilization and training times are shown in Table 9 and Figure 4. They clearly show a trade-off between training times(mins) and GPU Utilization(%). While our approach DWML had the lowest GPU utilization among all, the training time was reported the highest.

BLiMP Filtered On the BLiMP Filtered dataset, teacher-less methods demonstrate superior performance, with SD and DWML achieving 51.73% and 51.58% respectively, significantly outperforming their teacher-supervised counterparts KD (47.65%) and KD_DWML (47.47%). Among all approaches, our DWML framework shows strong performance, ranking second only to SD with a marginal difference of 0.15%. Notably, DWML substantially outperforms traditional KD by 3.93% and DML by 4.14%, validating the effectiveness of our dynamic weighting strategy in the absence of teacher supervision. Compared to the RoBERTa-base baseline (49.62%), both teacher-less methods show clear improvements, with DWML achieving a 1.96% gain, suggesting that peer learning alone can enhance syntactic understanding.

BLiMP Supplement The BLiMP Supplement results further reinforce the advantage of teacher-less methods, with SD achieving the highest score of 56.53%. Our DWML method (52.25%) outperforms DML (45.19%) by a substantial margin of 7.06%, though it falls behind SD. While KD (55.82%) and KD_DWML (53.65%) show competitive performance, the superior performance of SD demonstrates that teacher supervision isn't necessary for strong syntactic understanding. All distillation methods except DML surpass the RoBERTa-base baseline (48.9%) by a significant margin, with our DWML showing a 3.35% improvement, further validating the effectiveness of peer learning for syntactic tasks.

BLiMP Filtered						
Method	Teacher	Peer 1 ($60M$)	Peer 2 ($42M$)	Peer 3 ($34M$)	Peer 4 ($28M$)	Best↑
RoBERTa-base-125M	-	-	-	-	-	49.62
SD	No	51.73	50.04	50.31	51.18	**51.73**
KD	Yes	46.47	47.25	47.09	47.65	47.65
DML	No	47.01	47.77	47.21	47.16	47.44
KD_DWML (Ours)	Yes	47.05	47.28	47.47	46.66	47.47
DWML (Ours)	No	50.45	51.58	51.46	50.63	51.58

BLiMP Supplement						
Method	Teacher	Peer 1 ($60M$)	Peer 2 ($42M$)	Peer 3 ($34M$)	Peer 4 ($28M$)	Best↑
RoBERTa-base-125M	-	-	-	-	-	48.9
SD	No	53.03	54.78	49.63	56.53	**56.53**
KD	Yes	53.73	52.64	52.58	55.82	55.82
DML	No	44.74	45.14	45.19	44.96	45.19
KD_DWML (Ours)	Yes	52.21	53.09	53.34	53.65	53.65
DWML (Ours)	No	52.25	48.99	48.43	47.99	52.25

EWoK Filtered						
Method	Teacher	Peer 1 ($60M$)	Peer 2 ($42M$)	Peer 3 ($34M$)	Peer 4 ($28M$)	Best↑
RoBERTa-base-125M	-	-	-	-	-	51.6
SD	No	48.4	49.38	50.36	49.19	50.36
KD	Yes	50.12	50.3	51.56	50.42	51.56
DML	No	50.05	50.12	50.06	48.82	50.12
KD_DWML (Ours)	Yes	55.44	40.36	50.75	49.83	**55.44**
DWML (Ours)	No	49.98	49.84	49.08	50.29	50.29

Table 2: BLiMP Filtered, BLiMP Supplement, and EWoK scores for Text-only 10M dataset, comparing different distillation methods. Best accuracy scores (higher is better) are shown.

EWoK Filtered On the EWoK Filtered dataset, we observe a unique pattern where KD_DWML achieves the highest performance (55.44%), though teacher-less methods still show strong consistency, with SD, DML, and DWML achieving 50.36%, 50.12%, and 50.29% respectively. Interestingly, teacher-less methods perform slightly below the baseline, with a performance gap of up to 1.24%. This deviation from the pattern observed in BLiMP datasets suggests that world knowledge tasks may benefit more from teacher guidance, which could explain why KD_DWML achieved the best performance with a substantial 3.84% improvement over the baseline. This finding indicates that while peer learning is effective for syntactic tasks, world knowledge acquisition might require the structured guidance that teacher supervision provides.

5.3 Ablation studies

We compare the following modifications to the original DWML architecture :

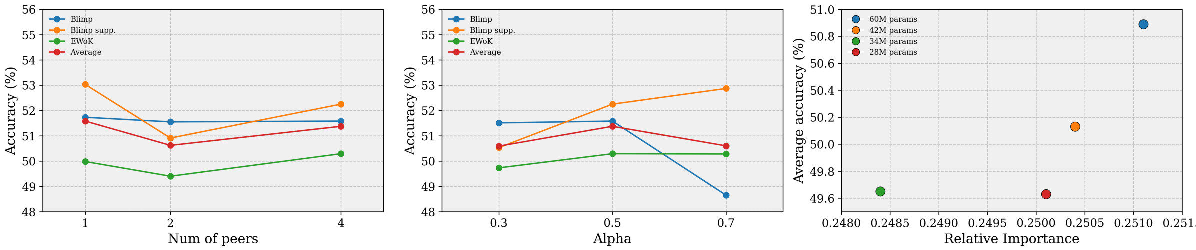
1. **Varying number of students** : The effect of using different number of student networks during training.

2. **Varying α ratio between label and peer supervision** : The effect of using different α in equation 2 that balances KL divergence and cross-entropy loss.

3. **Effect of dynamic student weights** : Determining if learning peer weights during training affect model performance (average accuracy %)

4. **Effect of model size** : Determining if model sizes affected model performance (average accuracy %)

Effect of Varying number of student models Figure 2(a) illustrates the impact of increasing the number of peer networks in our DWML framework. Performance on syntactic tasks, as measured by BLiMP and BLiMP Supplemental, shows modest variations across different peer counts. For BLiMP, we observe a slight decrease from 1 to 2 peers (51.73% to 51.55%), followed by a slight increase with 4 peers (51.58%). BLiMP Supplemental shows more variation, starting at 53.03%, dropping to 50.91% with two peers, and then increasing to 52.25% with four peers. The average performance across these metrics shows a similar pattern, starting at 51.58% with one peer, decreasing to 50.62% with two peers, and slightly recovering to 51.37% with four peers. These results indicate that while increasing the number of peers does



(a) Effect of number of peers

(b) Effect of alpha

(c) Effect of dynamic relative student importance and model size

Figure 2: Performance comparison across different experimental settings for 10M dataset: (left) varying number of peers, showing how model performance changes with different peer counts; (middle) impact of alpha parameter in the loss function on model accuracy; (right) relationship between relative importance and accuracy for different model sizes.

affect performance, the differences are relatively small, with no clear advantage for any particular peer configuration. This suggests that adding more peers may not necessarily lead to substantial gains in syntactic understanding tasks.

Effect of Varying Alpha Figure 2(b) demonstrates the impact of varying the alpha parameter, which balances the trade-off between cross-entropy loss and peer knowledge distillation in our loss function (Equation 2). With $\alpha = 0.3$, indicating stronger emphasis on label supervision, we observe the lowest performance. At $\alpha = 0.5$, representing an equal balance between label supervision and peer knowledge, performance improves across all metrics. However, when $\alpha = 0.7$, shifting focus more towards peer knowledge, we see mixed results with a notable decline in BLiMP (48.65%) while BLiMP Supplemental shows improvement (52.87%). This pattern suggests that $\alpha = 0.5$ provides an optimal balance: when α is too low (0.3), the models don't fully leverage peer knowledge, and when too high (0.7), excessive reliance on peer learning may compromise individual model performance. The results empirically validate our choice of $\alpha = 0.5$ as a balanced configuration for our DWML framework.

Effect of Dynamic Relative Student Importance Figure 2(c) reveals a positive correlation between dynamically learned importance weights and model performance ($R = 0.7$). Models with higher importance weights demonstrate better accuracy, as shown by the 60M parameter model achieving 50.89% accuracy with a 0.2511 weight, compared to the 28M model's 49.63% accuracy with a 0.2484 weight. This near-perfect linear relationship between assigned weights and performance validates our bi-level optimization approach, confirming that the framework successfully identifies and assigns higher weights to more capable models.

Effect of Model Size Figure 2(c) shows that model performance generally increases with model size, with the 60M parameter model achieving 50.89% accuracy, followed by 50.13% for 42M, 49.65% for 34M, and 49.63% for 28M parameters. This positive correlation between model size and performance aligns with previous findings, including those from the Chinchilla study (Hoffmann et al., 2024).

6 Conclusion

In this paper, we introduced Diversity Induced Weighted Mutual Learning (DWML) as an alternative to teacher-supervised knowledge distillation. While our approach showed modest improvements over the RoBERTa-base baseline, it was the simpler Self-Distillation method that achieved the strongest performance. Our ablation studies on our approach (DWML) revealed that two-peer configurations offered optimal efficiency, a balanced loss function ($\alpha = 0.5$) was crucial, and model performance correlated strongly with both dynamically learned importance weights and model size. Regarding computational efficiency, while DWML showed the lowest average GPU utilization, it required longer training times. Hence, in answering our research question about whether student knowledge sharing can match teacher-guided distillation on small datasets, we found that teacher-less methods can indeed match or exceed teacher-supervised approaches, but not necessarily through complex peer learning mechanisms. The success of simpler methods like SD suggests that the field might benefit from focusing on refined single-model approaches rather than elaborate multi-model frameworks. Future work should investigate why simpler teacher-less methods outperform more complex peer learning approaches, explore better neural architecture search techniques, and develop methods to reduce training time while maintaining low resource utilization.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The AMARA corpus: Building parallel language resources for the educational domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1856–1862, Reykjavik, Iceland. European Language Resources Association (ELRA).

Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. 2020. Online Knowledge Distillation with Diverse Peers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3430–3437.

Leshem Choshen, Ryan Cotterell, Michael Y Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. [Call for Papers] The 2nd BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2015. BinaryConnect: training deep neural networks with binary weights during propagations. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3123–3131, Cambridge, MA, USA. MIT Press.

Mathias Creutz. 2018. Open subtitles paraphrase corpus for six languages. *arXiv preprint arXiv:1809.06142*.

Shangchen Du, Shan You, Xiaojie Li, Jianlong Wu, Fei Wang, Chen Qian, and Changshui Zhang. 2020. Agree to Disagree: Adaptive Ensemble Knowledge Distillation in Gradient Space. In *Advances in Neural Information Processing Systems*, volume 33, pages 12345–12355. Curran Associates, Inc.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both weights and connections for efficient neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1135–1143, Cambridge, MA, USA. MIT Press.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

Geoffrey E. Hinton, O. Vinyals, and J. Dean. 2015. Distilling the Knowledge in a Neural Network. *ArXiv*.

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, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. 2024. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 30016–30030, Red Hook, NY, USA. Curran Associates Inc.

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 Akyürek, 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 [cs] version: 1.

Kirthevasan Kandasamy, Willie Neiswanger, Jeff Schneider, Barnabas Poczos, and Eric P Xing. 2018. Neural Architecture Search with Bayesian Optimisation and Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Xiaojie Li, Jianlong Wu, Hongyu Fang, Yue Liao, Fei Wang, and Chen Qian. 2020. Local Correlation Consistency for Knowledge Distillation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12357, pages 18–33. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint*. ArXiv:1907.11692 [cs].

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. 2017. Learning Efficient Convolutional Networks through Network Slimming. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2755–2763, Venice. IEEE.

Brian MacWhinney. 2000. *The CHILDES project: Tools for analyzing talk: Transcription format and programs, Vol. 1, 3rd ed.* The CHILDES project: Tools for analyzing talk: Transcription format and programs, Vol. 1, 3rd ed. Lawrence Erlbaum Associates Publishers, Mahwah, NJ, US. Pages: xi, 366.

Fernando Nogueira. 2014–. Bayesian Optimization: Open source constrained global optimization tool for Python.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. 2021. A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions. *ACM Comput. Surv.*, 54(4):76:1–76:34.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

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. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. 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. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Luting Wang, Xiaojie Li, Yue Liao, Zeren Jiang, Jianlong Wu, Fei Wang, Chen Qian, and Si Liu. 2022. HEAD: HEtero-Assists Distillation for Heterogeneous Object Detectors.

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. Place: Cambridge, MA Publisher: MIT Press.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. *Preprint*, arXiv:1905.08094.

Miao Zhang, Li Wang, David Campos, Wei Huang, Chenjuan Guo, and Bin Yang. Weighted Mutual Learning with Diversity-Driven Model Compression.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. 2018. Deep Mutual Learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, Salt Lake City, UT. IEEE.

A Pretraining Hyperparameters

Hyperparameters	Base	4 peer models				2 peer models		1 peer model
		1*	2	3**	4	1	2	
Number of parameters	125M	60M	42M	34M	28M	60M	42M	60M
Number of layers	12	8	16	32	8	8	16	8
Hidden size	768	512	256	128	256	512	256	512
FF intermediate size	3072	3072	3072	3072	3072	3072	3072	3072
Vocabulary size	50265	50265	50265	50265	50265	50265	50265	50265
Attention heads	12	32	8	4	8	32	8	32
Hidden dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Attention dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Training steps	150	150	150	150	150	150	150	150
Mini batch size	3	3	3	3	3	3	3	3
Num. of mini batches	60	60	60	60	60	60	60	60
Sequence length	514	514	514	514	514	514	514	514
Warmup ratio	0.03%	0.03%	0.03%	0.03%	0.03%	0.03%	0.03%	0.03%
Initial learning rate	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Final learning rate	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
Learning rate scheduler	cosine	cosine	cosine	cosine	cosine	cosine	cosine	cosine
Weight decay	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Layer norm ϵ	1.00E-12	1.00E-12	1.00E-12	1.00E-12	1.00E-12	1.00E-12	1.00E-12	1.00E-12
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
β_1	0.9	0.9	0.9	0.9	0.9	0.9	0.9	0.9
β_2	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
Gradient clipping	1	1	1	1	1	1	1	1

*Selected for 10M dataset. **Selected for 100M dataset.

Table 3: Pre-training hyperparameters for base and 4, 2 and 1 peer models for the DWML framework. The same set of hyperparameters are used for other distillation methods for an apple-to-apple comparison.

B Finetuning Hyperparameters

Hyperparameters	**Full fine-tuning**
Random seed	643
Batch size	32
Number of epochs	6
Dropout	0.1
Peak learning rate	2.50E-06
Learning rate decay	cosine
Weight decay	0.1
Optimizer	AdamW
Adam β_1	0.9
Adam β_2	0.999
Warmup steps	3

Table 4: Hyperparameters for full fine-tuning the GLUE, SuperGLUE task. We use the same fine-tuning script for comparison of RoBERTa-base and our DWML models.

C Dataset

Dataset	Domain	# Words STRICT-SMALL	# Words STRICT	Proportion
CHILDES MacWhinney (2000)	Child-directed speech	0.44M	4.21M	5%
British National Corpus (BNC),[1] dialogue portion	Dialogue	0.86M	8.16M	8%
Children's Book Test Hill et al. (2015)	Children's books	0.57M	5.55M	6%
Children's Stories Text Corpus[2]	Children's books	0.34M	3.22M	3%
Standardized Project Gutenberg Corpus Gerlach and Font-Clos (2020)	Written English	0.99M	9.46M	10%
OpenSubtitles Creutz (2018)	Movie subtitles	3.09M	31.28M	31%
QCRI Educational Domain Corpus (QED; Abdelali et al., 2014)	Educational video subtitles	1.04M	10.24M	11%
Wikipedia[3]	Wikipedia (English)	0.99M	10.08M	10%
Simple Wikipedia[4]	Wikipedia (Simple English)	1.52M	14.66M	15%
Switchboard Dialog Act Corpus (Stolcke et al., 2000)	Dialogue	0.12M	1.18M	1%
Total	–	9.96M	98.04M	100%

Table 5: The contents of datasets for the the 10M and 100M tracks; the table is taken from . [1]http://www.natcorp.ox.ac.uk [2]https://www.kaggle.com/datasets/edenbd/children-stories-text-corpus [3]https://dumps.wikimedia.org/enwiki/20221220/ [4]https://dumps.wikimedia.org/simplewiki/20221201/

D Detailed results

D.1 BLiMP

Method	AA	AS	B	CR	DNA	E	FG	IF	IFS	NPI	Q	SVA	AVG
RoBERTa_KD_DWML_peer1	39.60	50.40	48.70	53.70	48.50	45.10	39.10	38.10	46.90	44.40	46.80	51.10	46.00
RoBERTa_KD_DWML_peer2	39.50	50.30	52.60	53.70	48.40	49.40	36.70	33.20	47.10	45.50	46.30	51.20	46.20
RoBERTa_KD_DWML_peer3	39.40	50.30	53.60	53.70	48.30	51.40	36.70	33.30	47.80	45.10	46.30	51.30	46.40
RoBERTa_KD_DWML_peer4	39.80	50.60	50.10	53.10	48.70	43.60	37.30	27.50	48.20	44.30	46.10	49.90	44.90
RoBERTa_KD_peer1	39.60	49.50	50.30	51.90	49.40	46.80	37.10	40.10	46.00	40.30	48.00	50.80	45.80
RoBERTa_KD_peer2	39.60	50.30	53.30	**53.90**	48.50	48.60	36.50	32.70	47.20	44.70	45.90	51.20	46.10
RoBERTa_KD_peer3	39.70	50.30	51.80	53.30	48.40	49.10	36.60	33.20	47.60	44.20	46.70	51.10	46.00
RoBERTa_KD_peer4	54.30	49.90	52.30	56.70	45.80	48.70	37.70	32.20	48.90	44.10	49.50	48.40	47.40
RoBERTa_SD_peer1	**59.10**	**51.90**	48.60	47.40	**54.10**	56.90	36.50	**53.00**	48.00	**66.20**	**60.70**	51.20	**52.80**
RoBERTa_SD_peer2	45.70	53.60	58.10	53.00	51.00	52.20	37.00	52.30	47.60	53.00	38.20	50.40	50.20
RoBERTa_SD_peer3	53.50	50.10	50.50	52.30	50.50	51.70	**61.80**	33.70	**57.10**	42.10	36.70	49.70	49.10
RoBERTa_SD_peer4	59.10	51.40	47.70	44.90	48.30	53.10	51.30	55.70	58.30	49.40	54.80	48.80	51.90
RoBERTa_base	38.90	47.90	**62.80**	49.70	48.60	48.40	27.50	53.40	55.00	49.90	60.40	51.40	49.50
DWML_2model_peer1	45.30	51.70	57.90	48.90	47.50	50.00	46.70	45.70	58.80	43.90	55.10	50.70	50.20
DWML_2model_peer2	45.30	51.70	57.70	48.90	47.50	50.10	46.60	45.70	59.50	43.80	54.90	50.70	50.20
DWML_4model_peer1	53.70	51.80	42.50	50.40	50.00	49.30	45.30	53.70	50.40	45.70	56.90	50.50	50.00
DWML_4model_peer2	53.90	51.80	42.70	50.60	50.00	49.80	45.30	53.60	50.60	50.40	57.10	50.60	50.50
DWML_4model_peer3	53.60	51.70	42.00	50.60	50.00	49.70	45.20	53.50	50.60	45.40	57.10	50.60	50.00
DWML_4model_peer4	53.80	51.60	42.50	50.30	50.00	49.80	45.20	53.60	50.10	50.90	57.20	50.50	50.50
DWML_alpha_3peer1	49.20	50.40	48.50	49.80	50.60	50.00	53.20	51.60	50.00	64.20	44.70	51.80	51.20
DWML_alpha_3peer2	48.90	50.60	47.90	49.70	50.40	50.40	53.10	52.10	50.10	64.00	44.50	51.70	51.10
DWML_alpha_3peer3	49.30	50.50	49.60	50.00	50.60	49.90	53.00	52.10	49.50	58.30	44.50	51.60	50.70
DWML_alpha_3peer4	49.20	50.30	48.20	49.80	50.60	50.00	53.30	51.50	49.60	63.50	44.70	**51.90**	51.00
DWML_alpha_7peer1	58.30	49.00	40.50	49.30	52.70	54.60	50.40	56.80	43.20	41.30	54.50	49.60	50.00
DWML_alpha_7peer2	58.60	49.20	40.60	49.80	52.70	54.40	50.40	57.10	43.80	41.90	57.70	49.80	50.50
DWML_alpha_7peer3	58.40	49.00	39.80	49.10	52.80	54.50	50.20	56.80	44.00	41.90	54.30	49.70	50.00
DWML_alpha_7peer4	58.40	49.10	40.00	49.20	52.70	**54.80**	50.20	56.80	43.90	42.30	58.60	49.70	50.50
DML_peer1	54.10	49.20	52.00	50.30	48.30	47.00	42.40	47.10	54.00	27.20	48.10	49.40	47.40
DML_peer2	53.90	49.20	54.90	50.30	48.40	46.40	42.60	46.60	53.90	32.10	48.00	49.20	48.00
DML_peer3	54.00	49.10	54.70	50.60	48.40	46.90	42.50	46.70	53.80	26.90	48.00	49.00	47.60
DML_peer4	54.10	49.10	54.60	50.30	48.30	46.70	42.60	46.70	53.70	26.60	48.10	49.30	47.50

Table 6: BLiMP results for models trained using different methods. The **bold** results represent the best model for each task. The metric used is accuracy (%). Acronyms: AA (Anaphor Agreement), AS (Argument Structure), B (Binding), CR (Control/Raising), DNA (Determiner-Noun Agreement), E (Ellipsis), FG (Filler-Gap), IF (Irregular Forms), IFS (Island Effects), NPI (NPI Licensing), Q (Quantifiers), SVA (Subject-verb agreement)

D.2 BLiMP Supplement

Method	subject_aux_ inversion	qa_congruence_ tricky	turn_ taking	hypernym	qa_congruence_ easy	average
KD_DWML_peer1	44.53	60.61	56.07	52.97	46.88	52.21
KD_DWML_peer2	50.61	58.79	56.07	53.09	46.88	53.09
KD_DWML_peer3	50.32	58.79	56.07	54.63	46.88	53.34
KD_DWML_peer4	49.96	58.18	55.71	52.85	51.56	53.65
KD_peer1	53.19	59.39	55.36	52.26	48.44	53.73
KD_peer2	48.00	58.18	55.71	52.85	48.44	52.64
KD_peer3	48.69	59.39	56.07	53.44	45.31	52.58
KD_peer4	55.50	62.42	55.36	54.28	51.56	55.82
SD_peer1	65.48	47.88	45.00	**56.77**	50.00	53.03
SD_peer2	54.20	47.88	51.79	52.85	**67.19**	54.78
SD_peer3	58.81	52.12	50.71	53.68	32.81	49.63
SD_peer4	66.12	59.39	52.86	54.28	50.00	**56.53**
DWML_2peer_1	42.40	**65.50**	45.40	49.90	54.70	51.50
DWML_2peer_2	42.00	**65.50**	46.40	50.40	51.60	51.20
DWML_alpha_3peer_1	63.00	50.30	44.30	49.40	45.30	50.50
DWML_alpha_3peer_2	63.10	50.90	46.10	48.80	43.80	50.50
DWML_alpha_3peer_3	60.00	50.30	44.60	50.10	42.20	49.40
DWML_alpha_3peer_4	62.90	50.30	45.00	50.70	43.80	50.50
DWML_alpha_7peer_1	69.70	50.90	57.50	49.60	31.30	51.80
DWML_alpha_7peer_2	70.30	52.70	57.50	51.10	32.80	52.90
DWML_alpha_7peer_3	**70.90**	50.30	57.90	51.00	31.30	52.20
DWML_alpha_7peer_4	70.50	52.10	**59.30**	51.20	32.80	53.20
RoBERTa_base	54.00	41.20	52.90	51.30	45.30	48.90
DML_peer_1	42.60	55.80	51.40	48.90	25.00	44.70
DML_peer_2	45.30	55.80	51.10	48.60	25.00	45.10
DML_peer_3	42.10	57.00	51.40	50.50	25.00	45.20
DML_peer_4	43.80	55.20	51.40	49.40	25.00	45.00
DWML_4peer_1	53.6	53.4	43.2	54.4	56.6	52.25
DWML_4peer_2	50.6	49.8	40.2	51.1	53.3	48.99
DWML_4peer_3	50.6	48.4	39.5	50.4	53.1	48.43
DWML_4peer_4	48.8	48.0	40.0	49.6	53.6	47.99

Table 7: Supplement BLiMP results for RoBERTa models trained using different distillation methods. All values are presented as percentages.The **bold** results represent the best model for each task.

D.3 EWoK

Method	SP	QP	PR	SI	PI	MP	MD	PD	AP	SR	AVG
RoBERTa_base	53.5	54.5	51.0	48.0	51.8	53.5	50.1	54.2	49.5	50.4	51.6
KD_peer_1	60.2	58.0	51.8	45.6	45.5	48.2	**52.7**	32.5	45.9	50.2	50.1
KD_peer_2	50.4	51.9	48.9	50.7	50.2	55.3	49.7	47.5	52.6	49.2	50.3
KD_peer_3	50.4	48.4	51.3	52.0	50.9	55.9	50.4	55.8	51.0	48.8	51.6
KD_peer_4	38.8	43.9	50.0	**60.5**	43.7	51.2	50.8	50.0	52.6	49.4	50.4
KD_DWML_peer_1	**63.3**	**62.4**	**53.8**	48.3	52.5	**68.2**	47.1	**61.7**	**53.8**	51.0	**55.4**
KD_DWML_peer_2	50.0	45.5	42.3	40.1	41.2	24.7	47.9	28.3	44.4	46.9	40.4
KD_DWML_peer_3	49.8	53.2	50.4	53.7	50.4	55.9	48.8	50.8	48.2	49.0	50.8
KD_DWML_peer_4	40.4	49.4	44.5	45.2	**54.1**	57.6	49.0	40.8	**52.0**	52.0	49.8
SD_peer_1	52.4	46.2	50.2	48.3	48.2	44.1	49.7	50.0	49.4	50.5	48.4
SD_peer_2	52.4	47.5	49.3	47.3	51.6	45.9	50.1	54.2	49.6	50.2	49.4
SD_peer_3	52.0	49.4	51.2	48.3	51.1	52.4	49.0	43.3	50.4	50.9	50.4
SD_peer_4	53.7	50.3	49.6	47.3	49.1	51.2	48.8	42.5	50.6	49.5	49.2
DWML_2peer_1	49.8	47.5	51.6	47.6	49.6	45.9	50.3	48.3	51.0	51.0	49.4
DWML_2peer_2	50.2	50.6	49.1	46.9	53.1	50.6	51.3	45.0	51.0	49.9	49.9
DWML_alpha_3peer_1	53.5	47.8	49.0	49.0	50.4	48.8	49.6	50.0	49.4	50.3	49.5
DWML_alpha_3peer_2	51.8	49.0	49.8	46.9	50.4	50.6	49.2	49.2	50.0	50.1	49.7
DWML_alpha_3peer_3	50.6	48.4	50.2	47.3	48.7	51.8	50.5	44.2	49.2	50.2	49.2
DWML_alpha_3peer_4	51.4	47.5	49.9	46.6	50.4	51.2	50.8	48.3	49.9	49.8	49.4
DWML_alpha_7peer_1	50.4	54.1	50.1	50.7	49.1	55.9	50.6	49.2	50.9	51.0	51.4
DWML_alpha_7peer_2	51.0	49.4	49.3	51.4	50.7	51.2	50.3	48.3	50.3	49.2	50.3
DWML_alpha_7peer_3	50.0	48.4	49.3	49.0	51.6	45.3	48.4	50.0	50.1	49.9	49.6
DWML_alpha_7peer_4	50.6	49.0	49.9	49.3	50.2	51.2	48.8	49.2	49.0	49.9	49.6
DML_peer_1	51.4	47.1	50.0	50.0	50.5	55.9	49.1	53.3	49.2	51.0	50.1
DML_peer_2	51.0	49.0	49.1	49.0	50.7	54.7	49.2	52.5	49.3	50.6	50.1
DML_peer_3	49.8	51.0	49.1	52.7	46.9	52.9	51.6	44.2	49.5	50.8	50.1
DML_peer_4	52.7	52.5	48.4	50.3	46.8	42.4	50.6	44.2	50.0	49.7	48.8
DWML_4model_peer_1	52.0	49.7	50.0	47.6	50.9	48.8	49.5	50.0	50.1	50.3	50.0
DWML_4model_peer_2	51.2	48.7	49.0	51.4	48.9	50.0	50.6	48.3	49.0	49.9	49.8
DWML_4model_peer_3	49.4	51.3	50.4	49.7	49.5	50.6	50.1	39.2	50.0	49.6	49.1
DWML_4model_peer_4	53.9	48.7	46.8	45.2	46.8	50.6	51.4	56.7	49.7	49.2	50.3

Table 8: EWOK evaluation results for different distillation methods. The **bold** results represent the best performance for each metric. Acronyms: SI (Social Interactions), SP (Social Properties), SR (Social Relations), PI (Physical Interactions), PD (Physical Dynamics), PR (Physical Relations), MD (Material Dynamics), MP (Material Properties), AP (Agent Properties), QP (Quantitative Properties). The metric used is accuracy, and results are presented as percentage values.The **bold** results represent the best model for each task.

E Peer importance training during distillation

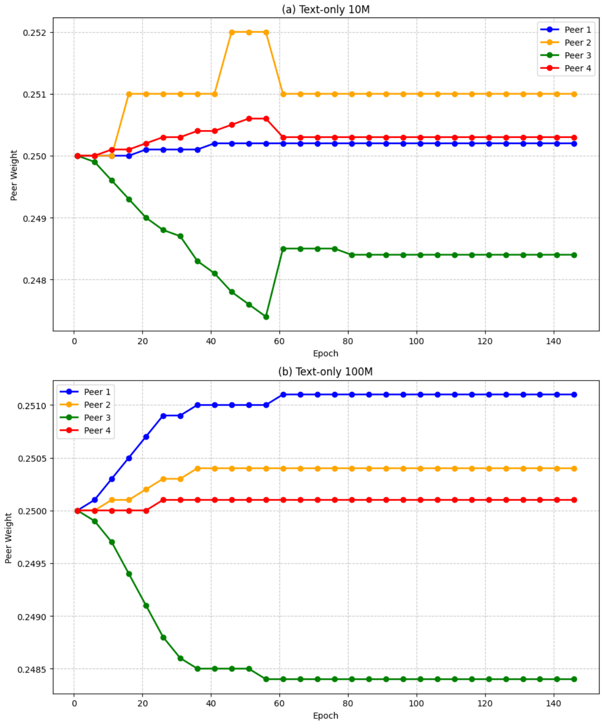


Figure 3: Peer importance weights dynamically trained using mirror descent algorithm as described in Equation 4

F GPU Utilization

Method	Max↓	Average↓	Training Time(mins)↓
RoBERTa_SD_10M_n_peer_4_4	56.07	52.92	**2.91**
RoBERTa_SD_10M_n_peer_4_3	**50.67**	48.47	8.51
RoBERTa_SD_10M_n_peer_4_2	55.60	51.10	4.98
RoBERTa_SD_10M_n_peer_4_1	73.07	68.47	3.02
RoBERTa_KD_DWML_10M_n_peer_4	69.80	62.71	25.51
RoBERTa_KD_10M_n_peer_4_4	63.40	58.68	3.53
RoBERTa_KD_10M_n_peer_4_3	53.93	51.48	9.03
RoBERTa_KD_10M_n_peer_4_2	60.93	56.09	5.52
RoBERTa_KD_10M_n_peer_4_1	78.00	63.89	4.01
RoBERTa_DWML_10M_n_peer_4	68.47	**43.20**	32.02
RoBERTa_DML_n_peer_4	86.8	43.66	8

Table 9: GPU utilization and training time for various RoBERTa distillation techniques (Lower is better). The RoBERTa_DML_n_peer_4 model shows the highest max utilization. In contrast, RoBERTa_SD_n_peer_4_1 maintains the highest average utilization (68.47%), indicating that training the largest peer model (60M) consistently increase GPU consumption. Our approach, DWML had the lowest average GPU utilisation over time, lower by $\tilde{2}0\%$ in comparison to its teacher-supervised counterpart (KD_DWML)

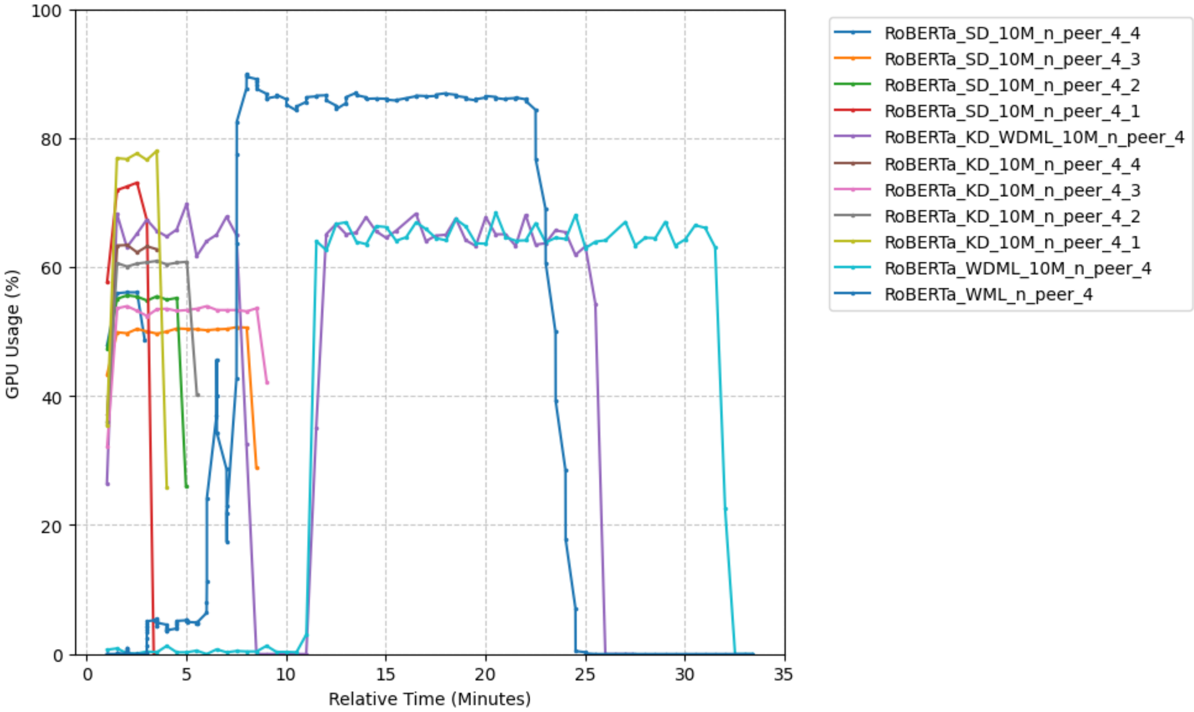


Figure 4: GPU utilization for different distillation methods.

Automatic Quality Estimation for Data Selection and Curriculum Learning

Hiep Nguyen and **Lynn Yip** and **Justin DeBenedetto**
Department of Computing Sciences
Villanova University

Abstract

The size of neural models within natural language processing has increased at a rapid pace in recent years. With this increase in model size comes an increase in the amount of training data required for training. While these larger models have shown strong performance, their use comes with added training and data costs, can be resource-prohibitive for many researchers, and uses an amount of language data that is not always available for all languages. This work focuses on exploring quality estimation as a method of data selection or filtering. The aim is to provide models with higher quality data as compared to larger amounts of data. This approach was applied to machine translation models with varying data sizes as well as to the BabyLM Challenge. Given the 100M word dataset provided in the BabyLM Challenge, we test out various strategies for selecting 10M words for pretraining and use a curriculum learning approach based on the quality estimation scoring. We find small improvements in certain data settings.

1 Introduction

In recent years, there has been a dramatic rise in the size of neural network models used for natural language processing tasks. To train these larger models, there has been a similar rise in the size of datasets used for training or pretraining. While these models have been quite successful, this trend comes with several downsides including the cost of creating these larger systems which also inhibits the ability of many researchers who lack access to the large scale computing resources required. By contrast, human language development occurs in children with exposure to far fewer words of training data. Inspired by this, the BabyLM Challenge (Choshen et al., 2024) focuses on "sample-efficient pretraining on a developmentally plausible corpus."

One approach to improve model performance in data-limited settings is curriculum learning (Elman, 1993). Just as human language learners are typically exposed to simpler language before building up to more complex utterances, curriculum learning involves increasing the difficulty of training examples over the course of model training. In order to do this, there must be some measure of "difficulty" in order to assign an ordering to training examples. In this work, we apply quality estimation scoring as an estimation of difficulty. These scores are used to train models for the BabyLM Challenge, specifically restricted to 10 million words or less of training data.

Quality estimation (QE) in machine translation scores the quality of translation output without the need for a reference translation (Specia et al., 2018). Through a series of experiments, we explore the effects of using QE to filter data for machine translation systems for both initial model training and fine-tuning, as well as the result of training on different quantities of data for each model (see Section 3). Prior work has shown that data filtering through QE can increase model performance (Batheja and Bhattacharyya, 2023). We explore that further in this work for both machine translation and language modeling in data restricted settings.

Since quality estimation scores the quality of the output of a machine translation system, it is likely that higher QE scores correspond to sentences which the system has an easier time translating. Motivated by this, we experiment with using QE scores as an estimation of the difficulty of a given sentence for an NLP system. In particular, we use this for data selection as well as for difficulty scoring for curriculum learning training of a "baby" language models as part of the 2024 BabyLM Challenge (see Section 4).

2 Related work

2.1 BabyLM Challenge and Curriculum Learning

As this is the second year of the BabyLM Challenge, there is a body of existing work which relates directly to our BabyLM experiments (Warstadt et al., 2023). There were many submissions (41.9% of teams) in last year's iteration which made use of curriculum learning. Using a curriculum to make training difficulty scale up during training is known

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 212–220
November 15-16, 2024

as curriculum learning (Bengio et al., 2009). It has been shown that reordering input data during training can have a large effect on model performance across tasks such as natural language inference (NLI) (Schluter and Varab, 2018) and neural machine translation (NMT) (Liu et al., 2020). The most similar approaches from last year's BabyLM Challenge to this current work were by Chobey et al. and by Hong et al.. In those works, a teacher language model was trained first and used to determine the curriculum for training a new model. We similarly are using another model to inform the curriculum, though the model and curriculum forming is done differently.

2.2 Quality Estimation

QE has been used to assist with both automated post-editing (APE) (Chatterjee et al., 2018) and human post-editing tasks (Béchara et al., 2021). QE can be used in tandem with APE to determine which sentences from a machine translation system need to be corrected (Chatterjee et al., 2018). In contrast, we use QE in this work to filter out data to be used for fine-tuning the machine translation model.

QE has also been used to extract high-quality data from both parallel and pseudo-parallel data for training machine translation systems (Batheja and Bhattacharyya, 2022, 2023). We take this work one step further by fine-tuning machine translation systems on the model's own output which was also filtered using QE. The results from fine-tuning on both high and low-quality data were evaluated.

3 Quality Estimation for Machine Translation

3.1 Methodology

3.1.1 Dataset

We used the German-English IWSLT 2017 dataset (Cettolo et al., 2017) for all machine translation experiments described in this section. The original dataset was initially divided into eight sets of different sizes ranging from approximately 1500 sentences to the full-sized set of 198669 sentences as shown in Table 1. The full dataset was first halved to create the next smallest sized dataset. This smaller dataset was then also halved to create the next smallest size and so on for all eight. Sentences that were removed during this process did not reappear in smaller sets. This ensured that each smaller set of sentences consisted solely of sentences from the larger set.

The smallest dataset split of roughly 1500 sentences was dropped due to the BLEU scores being too low to be meaningful after initial model training. All experiments listed were completed with the remaining seven splits of data.

3.1.2 Model Training

The fairseq (Ott et al., 2019) sequence modeling toolkit was used to train machine translation models from German to English. A new model was trained on each dataset split. The results from initial model training resulted in BLEU scores ranging from 0.04 to 36.82, with the largest dataset split corresponding to the highest BLEU score.

3.1.3 Quality Estimation Filtering

TransQuest is a framework for machine translation quality estimation that can be used to rate translations at either the word or sentence level (Ranasinghe et al., 2020). The SiameseTransQuest sentence-level quality estimation model was used throughout these experiments[1].

The quality estimation threshold to separate high-quality and low-quality sentences was determined by selecting the threshold that gave the widest range of filtered sentence quantity across all seven split datasets.

3.1.4 Model Fine-Tuning

Using the sentences that were filtered out using TransQuest quality estimation, the original fairseq translation models for the specified dataset split were fine-tuned on the filtered sentences. The BLEU scores were recorded after fine-tuning to see if any improvements had been made as a result of the fine-tuning. To replicate any of our results, please see our GitHub repository[2].

3.2 Experiments

Experiments 1 through 3 start with fairseq translation models trained on the original IWSLT 2017 German-English dataset splits. See Table 5 for the full results. In experiments 4 through 6, the original dataset is first filtered with QE and only the high-quality data is used for initial model training. QE is then used to filter the model output for fine-tuning (see Table 6).

3.2.1 Experiment 1

Seven fairseq models were trained on the original IWSLT 2017 German-English dataset which had been split into varying sizes. Each model was then used to translate the test set, which introduced new data to the model. The output translations went through TransQuest quality estimation. The low-quality sentences as rated by TransQuest were used to fine-tune the models.

For the models initially trained on the smallest splits, excluding the eighth split, fine-tuning resulted in BLEU score improvements from 0.46 and 0.76. The most significant score improvement was

[1] https://huggingface.co/TransQuest/siamesetransquest-da-en_de-wiki

[2] https://github.com/lsyip/mt-qe-filtering

Dataset Split	Number of Sentences	BLEU
1 (Full set)	198669	36.82
2	99335	33.15
3	49668	29.14
4	24834	23.00
5	12417	16.63
6	6209	11.75
7	3105	9.40
8	1553	0.04†

Table 1: Initial Model BLEU Scores for Experiments 1-3. Model trained on unfiltered data, finetuned on high or low quality data.
†Not used in experiments

seen in split 7, where the initial model had been trained on the smallest amount of data.

3.2.2 Experiment 2

The seven base translation models trained on the dataset splits remain the starting point for this experiment. This time, the models were used to translate the training sentences that they had been trained on, thus re-introducing the same data the model was trained on. The translation output went through TransQuest quality estimation and the low-quality sentences were used to fine-tune the models.

For this experiment, splits 5 and 6 had BLEU score improvements of over 0.6 points. The remaining splits did not show significant improvement after fine-tuning.

3.2.3 Experiment 3

Starting again with the seven base translation models, the models were again used to translate the training set. This translation output then went through TransQuest quality estimation and the high quality sentences were used for fine-tuning.

The highest BLEU score improvement for this pipeline was on split 5, which showed an increase of 0.41 points after fine-tuning on the high-quality sentences. The remaining splits did not show significant improvement in BLEU scores after finetuning.

3.2.4 Experiment 4

For this experiment, we first filtered each of the IWSLT 2017 splits through TransQuest quality estimation. See Table 2 for details. Next, new fairseq translation models were trained on the sentences that were rated to be of high quality. These models serve as the starting point for the following two pipelines. This setup mirrors the parallel corpus filtering via quality estimation previously done by Batheja and Bhattacharyya.

Dataset Split	Number of Sentences	BLEU
1	75898	30.56
2	36269	25.00
3	17776	19.13
4	8882	15.20
5	4286	6.90
6	2149	0.05
7	1082	0.07

Table 2: Initial Model BLEU Scores for Experiments 4-6. Model trained on filtered data, finetuned on high or low quality data.

3.2.5 Experiment 5

Using the new fairseq models that were trained in experiment 4, the model was asked to translate all sentences from their respective training set, which did not introduce new data to the model. The new translations were sent through TransQuest quality estimation and the sentences that were rated high-quality were used for fine-tuning.

After fine-tuning, the models trained on the smallest three splits did not show significant improvement to their BLEU scores. However, some improvement was made with the larger splits. The models initially trained on splits 1, 2, 3 and 4 showed BLEU score improvements of 0.38, 0.6, 0.58, and 0.46, respectively. It is important to note that with each larger split, the number of sentences in the fine-tuning set also increases as more translations were sent through quality estimation.

3.2.6 Experiment 6

Starting again with the fairseq models that were previously trained in experiment 4, these models were again used to translate sentences from their respective original training sets and the new translations were sent through TransQuest for quality estimation. The sentences that were rated to be of low-quality were then used to fine-tune the models.

After fine-tuning, smallest 3 splits did not show any improvements in BLEU score. Splits 1, 2, and 4 showed an increase in BLEU score between 0.21 and 0.28. Split 3 had the highest BLEU score increase of 0.51.

3.3 Machine Translation Results

In experiments 1-3, we observed that some models trained on smaller datasets saw improvements in BLEU score after fine-tuning on training data that had been filtered through quality estimation. The differences between using low-quality and high-quality data to fine-tune, however, were marginal. This suggests that the quality of the data may not matter as much as the quantity that is available. For the smaller datasets, improvements could be seen after fine-tuning in both the low and high-quality instances.

For experiments 4-6, which used TransQuest quality estimation to filter both the original dataset and the data for fine-tuning, the initial model BLEU scores were lower than the first three experiments due to having fewer training sentences. We observed that some improvements in BLEU score can be made after fine-tuning on the filtered high-quality on the larger dataset splits. The most significant differences after fine-tuning were seen in splits 2, 3, and 4.

4 BabyLM Challenge

4.1 Methodology

4.1.1 Dataset

The data for the BabyLM Challenge provided by the challenge organizers consists of text from six sources and was selected to represent language data that a human child may be exposed to when developing their language skills. The provided dataset contains 100 million words of text data. From this, we could form training datasets containing up to 10 million words to train models for the strict-small track.

4.1.2 Model and Training

The data preprocessing involved removing blank lines and special characters, with a focus on dialogue-related elements. The sentences within each dataset were then rearranged based on length to streamline the training process. After preprocessing, the sentences were translated from English to German using base translation models from fairseq. See Table 3 for metadata of processed datasets.

The quality of resulting pairs of German-English sentences was assessed using TransQuest and xCOMET frameworks. COMET, which stands for Crosslingual Optimized Metric for Evaluation of Translation, is a neural framework to predict human judgments on machine translation quality from source and target language samples (Rei et al., 2020). Specifically, the wmt23-cometkiwi-da-xl[3] model was chosen for xComet, and its results were compared to those from TransQuest. However, we found the scores from both models to be inconsistent with each other. In the end, the xComet scores were selected to rank and filter the data for training, since the TransQuest scores were heavily influenced by sentence length.

Our model is a RoBERTa (Liu et al., 2019) model. RoBERTa is a modification of the BERT (Devlin et al., 2018) model, which showed improved performance across several benchmarks.

We conducted several experiments to explore different strategies for selecting a training subset with a budget of 10 million words from the original 100 million words. The experiments varied based primarily on:

- The order of training: ascending or descending (original curriculum learning) order of quality estimation scores (equivalently, reversed machine comprehension level),
- The separation or combination of datasets from various sources,
- The number of hidden layers and heads during model training.

After filtering and training, models were evaluated using a standardized evaluation pipeline provided by the organizers to compute their scores on the BLiMP and EWoK benchmarks.

Code to train our models can be found on GitHub[4].

4.2 Experiments

4.2.1 Experiment 1

For this experiment, all sources were combined and rearranged based on their xComet scores. The 10 million words in sentences with highest scores were kept and divided into 3 files, namely easy (top 2 million), medium (next 4 million), and hard (next 4 million). Sentences with higher QE scores are considered to be easier sentences. Those files are trained in order from easy to hard, following the curriculum learning approach.

4.2.2 Experiments 2 and 3

For Experiment 2, 1.8 million words from the highest scored sentences of each sources were selected, except for Switchboard from which all 0.8 million words were taken. For Experiment 3, we did the opposite, by selecting the 1.8 million words from the lowest scored sentences from each source. This means we followed the typical order for curriculum learning in Experiment 2 and the reversed order in Experiment 3.

In both experiments, the following file order was used for training: CHILDES, OpenSubtitles, Switchboard, BNC_spoken, Simple_wiki, and Gutenberg.

4.2.3 Experiments 4-6

For Experiment 4, we tried to replicate Experiment 1 in the way word data were selected, starting by combining all sources into one stream for score ranking. However, we divided the word budget into 5 files grouped by QE score with each file containing around 2 million words. The model is trained on these files from easiest to hardest in Experiment 4 and reversed order in Experiment 6.

For Experiment 5, instead of selecting 10 million words from highest scored sentences, we filtered

[3] https://huggingface.co/Unbabel/wmt23-cometkiwi-da-xl

[4] https://github.com/jdebened/BabyLM2024

Dataset	Description	# Words (Original)	# Words (Processed)
CHILDES	Child-directed speech	28.9M	15.6M
British National Corpus (BNC)	Dialogue	7.7M	5.3M
Standardized Project Gutenberg Corpus	Written English	26.3M	21.7M
OpenSubtitles	Movie subtitles	20.0M	13.5M
Simple Wikipedia	Wikipedia (Simple English)	14.7M	11.5M
Switchboard Dialog Act Corpus	Dialogue	1.3M	0.9M
Total		**99M**	**68.5M**

Table 3: Original and Processed Dataset provided for the strict track of the BabyLM Challenge. Dataset names, domain descriptions, and word counts

those from lowest scored ones and trained resulting files in the order of hardest to easiest files.

4.2.4 Experiments 7-10

For these experiments, only one source was used in each experiment, namely either CHILDES or Gutenberg. In Experiments 7 and 8, the 10 million words were selected from lowest scores of CHILDES and Gutenberg datasets respectively. The order of training is from sentences with lowest scores to those with higher scores, opposite of expected curriculum learning order.

In Experiment 9, we filtered down to the 10 million words from highest scoring sentences of CHILDES to compare with the result from Experiment 7. It is noted that this comparison is based on data selection of highest and lowest scored sentences as well as training order of increasing and decreasing complexity.

In Experiment 10, we used the same subset of 10 million words from Experiment 10. However, the number of hidden layers and heads were doubled for further comparison.

The motivation behind choosing these sources rather than others is because we wanted to test the opposition between child-directed speech and written texts.

4.2.5 Experiments 11 and 12

For these experiments, we tried to replicate Experiments 4 and 5 respectively. However, we decided to split into smaller files, each with 1 million words.

4.2.6 Experiments 13 and 14

For these experiments, the mixture of 5 million words from highest scored sentences and 5 million words from lowest scored ones were used.

The primary difference between these experiments are based on their order of training. While Experiment 13's order followed curriculum training, Experiment 14 did the opposite.

Full experiment descriptions, mainly in how data was selected for model training, can be found at Table 4.

4.3 Results

Full experiment results including BLiMP and EWoK scores from the evaluation pipeline can be found at Table 4.

Evaluation pipeline provided by BabyLM Challenge 2024 included zero shot evaluation on tasks from the BLiMP benchmark and hidden evaluation tasks from the Ewok benchmark (Warstadt et al., 2020; Ivanova et al., 2024).

BLiMP is made up of tasks designed to test how well language models adhere to the structure of English. Each task presents a pair of sentences, where one is grammatically correct, and the other is incorrect, with the two sentences differing as little as possible. A model is considered accurate for a given example if it assigns a higher probability to the correct sentence in the pair (Warstadt et al., 2023).

Elements of World Knowledge (EWoK) framework evaluates world modeling in language models by testing their ability to use knowledge of concepts across physical and social domains to determine plausible or implausible contexts. It flexibly constructs multi-step scenarios, targets specific cognitive concepts, and generates controlled evaluation items using a template-based approach. This framework focuses on how well language models can productively apply concept knowledge, rather than just matching individual sentences or facts (Ivanova et al., 2024).

From the table of results, we found several patterns in the varied BLiMP and EWoK scores:

- Models with order of training from harder to easier, opposite to expected order from curriculum learning (decreasing complexity) had slightly higher BLiMP_complement scores compared to others with/without same datasets such as models 3, 5, 10, 12, 14. The exception in this case is Model 6, compared to Model 4. However, the BLiMP_filtered and EWoK_filtered scores did not experience the similar pattern with no noticeable improvement for any order. This inconsistency may stem from our assumption of the relationship

#	Data setup	BLiMP complement	BLiMP filtered	EWoK filtered	Details
1	All data sources Sources combined CL training order	58.01	60.69	49.99	10M highest QE scores separated into 2M highest, next 4M, next 4M
2	All data sources Sources kept separate CL training order	54.98	60.87	49.15	1.8M words of each source by highest QE score (Switchboard max 0.8M)
3	All data sources Sources kept separate Reversed CL order	60.25	60.17	50.47	1.8M words of each source by lowest QE score (Switchboard max 0.8M)
4	All data sources Sources combined CL training order	58.92	61.01	50.00	10M highest QE scores separated into 5 equal-sized files
5	All data sources Sources combined Reversed CL order	61.41	60.45	50.10	10M lowest QE scores separated into 5 equal-sized files
6	All data sources Sources combined Reversed CL order	56.83	60.31	49.71	10M highest QE scores separated into 5 equal-sized files, train in the reverse order (compared to experiment 4)
7	CHILDES data only Reversed CL order	59.34	57.80	50.21	10M lowest QE scores separated into 5 equal-sized files
8	Gutenberg data only Reversed CL order	58.27	61.94	50.46	10M lowest QE scores separated into 5 equal-sized files
9	CHILDES data only CL training order	55.41	57.99	50.30	10M highest QE scores separated into 5 equal-sized files
10	Gutenberg data only Reversed CL order	59.73	**62.39**	49.66	10M lowest QE scores separated into 5 equal-sized files, double the number of hidden layers and heads (compared to experiment 8)
11	All data sources Sources combined CL training order	56.92	61.29	**50.97**	10M highest QE scores separated into 10 equal-sized files
12	All data sources Sources combined Reversed CL order	59.80	60.11	50.55	10M lowest QE scores separated into 10 equal-sized files
13	All data sources Sources combined CL training order	59.74	60.35	50.31	5M highest QE scores separated into 5 equal-sized files, then 5M lowest QE scores separated into 5 equal-sized files
14	All data sources Sources combined Reversed CL order	**63.02**	60.66	50.18	5M highest QE scores separated into 5 equal-sized files, then 5M lowest QE scores separated into 5 equal-sized files; train in reverse order (compared to experiment 13)

Table 4: Experiments setups and results (%). Comparison between models trained on 10 million word budget filtered from original 100 million words provided in the 2024 BabyLM Challenge. **Bolded** values show best in column. Strategies to filter the data to form training datasets containing up to 10 million words to train models.

between quality estimation and machine comprehension level.

- Models with combined sources did not show superior results in BLiMP_complement scores compared to separated ones. Models with multiple sources also did not outperform single-source models. However, regarding the BLiMP_filtered scores, single-source model using Gutenberg showed better performance compared to multiple-source models or other single-source models, especially derived from CHILDES data. Additionally, this may also relate to the fact that Gutenberg's written style and higher quality can improve the performance.

- Models' performance and the number of files to train were not proportional in terms of BLiMP_complement scores, but showed a clear positive correlation in EWoK_filtered. Models using 5 training files get the highest BLiMP_complement in comparison to 3 or 10 files. Meanwhile, in the case of implementing the curriculum learning order (models 1, 4, 11), the BLiMP_filtered accuracy positively correlated with the number of files.

- Models using doubled number of heads and hidden layers took more time to train and had better BLiMP_complement and BLiMP_filtered scores (models 8 and 10), but not EWoK_filtered scores.

5 Conclusion

This work explored quality estimation for data filtering and curriculum learning on both machine translation systems and language models. As shown in our machine translation experiments (see Section 3.2), modest improvements can be obtained through finetuning on filtered data. This benefit largely went away as the data size scaled up to the full IWSLT17 dataset, suggesting that this method has more use for certain data limited settings rather than for general model use. Furthermore, base model performance went up more noticeably when additional data was added, showing that more data made a larger difference than higher quality data in this setting.

For the BabyLM Challenge strict-small track, teams could form datasets consisting of up to 10 million words to train their language models. We explored several options for data selection from the provided 100 million word dataset. Each model was then trained using a curriculum learning approach based on quality estimation scoring. Overall, data source made a bigger difference to model performance than curriculum choice. In particular, models trained using the Project Gutenberg data generally had higher scores on downstream tasks. This suggests that while the other data sources are useful for human children learning language, the higher quality data available in the Gutenberg dataset produced a better language model.

Acknowledgments

This work used the Augie High-Performance Computing cluster, funded by award NSF 2018933, at Villanova University.

References

Akshay Batheja and Pushpak Bhattacharyya. 2022. Improving machine translation with phrase pair injection and corpus filtering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5395–5400.

Akshay Batheja and Pushpak Bhattacharyya. 2023. A little is enough: Few-shot quality estimation based corpus filtering improves machine translation. In *The 61st Annual Meeting Of The Association For Computational Linguistics.*

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Hannah Béchara, Constantin Orăsan, Carla Parra Escartín, Marcos Zampieri, and William Lowe. 2021. The role of machine translation quality estimation in the post-editing workflow. *Informatics*, 8(3).

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan. International Workshop on Spoken Language Translation.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Frédéric Blain, and Lucia Specia. 2018. Combining quality estimation and automatic post-editing to enhance machine translation output. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 26–38, Boston, MA. Association for Machine Translation in the Americas.

Aryaman Chobey, Oliver Smith, Anzi Wang, and Grusha Prasad. 2023. Can training neural language models on a curriculum with developmentally plausible data improve alignment with human reading behavior? In *Proceedings of the BabyLM Challenge at the 27th Conference* *on Computational Natural Language Learning*, pages 98–111.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jeffrey L Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Xudong Hong, Sharid Loáiciga, and Asad Sayeed. 2023. A surprisal oracle for active curriculum language modeling. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 259–268.

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

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. Transquest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Natalie Schluter and Daniel Varab. 2018. When data permutations are pathological: the case of neural natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4935–4939, Brussels, Belgium. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón F. Astudillo, and André F. T. Martins. 2018. Findings of the WMT 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 689–709, Belgium, Brussels. Association for Computational Linguistics.

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.

A Appendix

			Experiment 1		Experiment 2		Experiment 3	
Split	Train Sents	Initial BLEU	FT† Sents	BLEU	FT Sents	BLEU	FT Sents	BLEU
1	198669	36.82	1884	36.81 (-0.01)	128246	36.94 (+0.12)	70423	36.84 (+0.02)
2	99335	33.15	1862	33.17 (+0.02)	66261	33.30 (+0.15)	33074	33.29 (+0.14)
3	49668	29.14	1962	29.43 (+0.29)	33540	29.45 (+0.31)	16128	29.49 (+0.35)
4	24834	23.00	2110	23.15 (+0.15)	16934	23.28 (+0.28)	7900	23.1 (+0.10)
5	12417	16.63	2232	**17.09 (+0.46)**	8670	**17.24 (+0.61)**	3747	**17.04 (+0.41)**
6	6209	11.75	2682	**12.21 (+0.46)**	4419	**12.41 (+0.66)**	1790	11.86 (+0.11)
7	3105	9.40	2928	**10.14 (+0.74)**	2185	9.42 (+0.02)	920	9.41 (+0.01)

Table 5: Experiment 1-3 Results. Model trained on unfiltered IWSLT17 dataset, fine-tuned on high or low quality data.
†Fine-tune

	Experiment 4		Experiment 5		Experiment 6	
Split	Train Sents	Initial BLEU	FT† Sents	BLEU	FT† Sents	BLEU
1	75898	30.56	60187	30.94 (+0.38)	15711	30.77 (+0.21)
2	36269	25.00	29657	25.6 (+0.60)	6612	25.28 (+0.28)
3	17776	19.13	14433	**19.71 (+0.58)**	3343	**19.64 (+0.51)**
4	8882	15.20	7728	**15.66 (+0.46)**	1154	15.44 (+0.24)
5	4286	6.90	1402	6.87 (-0.03)	2884	6.72 (-0.18)
6	2149	0.05	178	0.06 (+0.01)	1971	0.05
7	1082	0.07	83	0.07	999	0.07

Table 6: Experiment 4-6 Results. Model trained on filtered IWSLT17 dataset, fine-tuned on high or low quality data.
†Fine-tune

Using Curriculum Masking Based on Child Language Development to Train a Large Language Model with Limited Training Data

Evan Lucas[1], Dylan Gaines[1], Tagore Rao Kosireddy[1], Kevin Li[1], Timothy C. Havens[1]

[1]Michigan Technological University
1400 Townsend Drive
Houghton, Michigan, United States

Abstract

In this paper we detail our submissions to the STRICT and STRICT-SMALL tracks of the 2024 BabyLM Challenge. We approach this challenge with two methodologies: i) use of a novel dataset, and ii) development of a pre-training technique based on the fusion of child language acquisition with traditional masked language modeling, which we call *curriculum masking*. The novel dataset used for this task is based on user submissions to the Reddit forum (i.e., subreddit) "Explain Like I'm Five", which explains diverse concepts using simple language. Curriculum masking works by creating learning phases based on a standard child language development timeline, where the masked words learned by the model start with simple nouns and gradually expand to include more complex parts of speech. We show that using internet-based training data shows a small improvement in evaluation scores as compared to baseline training data. Our proposed pre-training method of curriculum masking is conceptually novel and also shows improved rates of learning over typical masked language modeling pre-training, potentially allowing for good performance with fewer total epochs on smaller training datasets. Code for the curriculum masking implementation is shared at https://github.com/evan-person/curriculumMaskingBabyLM2024.[1]

1 Introduction

Children acquire language skills through exposure to an estimated two to seven million words per year. However, contemporary large language models (LLMs) require training on massive datasets comprising billions to trillions of words to achieve similar linguistic capabilities. The vast disparity between human language acquisition and current machine learning practices can be shown from the Chinchilla model (Hoffmann et al., 2022), which was trained on 1.4 trillion words.

To address these disparities, the BabyLM Challenge was established to explore the feasibility of pre-training LLMs on datasets comparable in size to those encountered during early childhood language development. It continues on this mission, imposing strict limits on the size and composition of training datasets and aims to create models that learn language in a child-like manner.

In this paper, we present our submissions to both the STRICT and STRICT-SMALL of the 2024 BabyLM Challenge. We leverage a novel dataset, sourced from the Reddit forum (i.e. subreddit) *Explain Like I'm Five* (ELI5). We introduce a curriculum masking training strategy that we designed to mimic how children learn language. Traditional *Masked Language Modeling* (MLM) masks random words during training, which does not accurately reflect the structured manner in which children acquire language. Our curriculum masking approach organizes the process into a schedule of stages, starting with simpler words such as nouns, then gradually incorporating more complex words like adjectives and verbs. We hypothesized this scheduling method would help the model build a stronger foundation in language before tackling more advanced sentence structures. We do not count the added information of POS tags as additional word count, as it is only an additional categorical variable attached each token. In conversations on the challenge Slack channel, this was agreed to not count towards overall word count for this reason, and so although it is additional information, we make the claim that it does not count as increased training words. The following sections of this paper describe our dataset preparation, implementation of curriculum masking, experimental results, and discussion of the effectiveness of our approach.

[1]https://github.com/evan-person/curriculumMaskingBabyLM2024

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 221–228
November 15-16, 2024

2 Related Work

To understand the state of related work, we reviewed the most relevant papers from the BabyLM 2023 challenge (Warstadt et al., 2023) and performed searches for similar concepts to what we propose in this work. Several works in the prior challenge utilized curriculum learning, though they administered their curricula through an intentional sequencing of the examples in their training set as opposed to the curriculum-based masking approach we use.

Martinez et al. (2023) investigated curriculum learning strategies for language model pre-training using limited data. Similar to our curriculum masking, their approach progressively increased task complexity, structuring model training in stages. Although their methods did not consistently outperform non-curriculum baselines, their focus on vocabulary and data pacing offers valuable insights for optimizing training with limited resources.

DeBenedetto (2023) applied a curriculum learning strategy for low-resource settings, where datasets were ranked by difficulty using a bytes-per-line metric. Simpler datasets, such as spoken transcriptions, were introduced first, then more complex datasets were gradually introduced during training. Their approach outperformed baseline models in most downstream tasks, including BLiMP and SuperGLUE. Their curriculum learning methods demonstrated consistent improvements in performance, particularly when trained with more epochs.

Bunzeck and Zarrieß (2023) designed a curriculum learning approach based on child-directed speech which showed an improvement for certain tasks like anaphor agreement, irregular forms, and quantifiers. Their work, similar to ours, involved using curriculum learning focused on word frequency and sentence structure. However, they used a static data ordering approach where the training data was organized in a fixed sequence.

Curriculum masking as a concept has been applied successfully in other domains, such as computer vision. Jarca et al. (2024) developed a curriculum based masking strategy for vision tasks. They show that by using a curriculum-based masking approach for training vision models they are able to outperform the same model architecture on some common image classification tasks.

3 Method

For our submission, we make two primary contributions: i) a new ELI5 dataset, and ii) a method of curriculum learning that modifies the MLM pretraining task to mimic child language development. In this section we review these two contributions as well as the other method choices made.

3.1 Dataset

The curated dataset provided by the challenge organizers contains various sources of child and child-directed speech. In addition to this dataset, we created a novel dataset using the subreddit *Explain Like I'm Five*[2]. On this particular subreddit, users can pose questions on almost any topic. Other users' responses to these questions are required to be free of technical jargon and tend to use simplified concepts. While the responses are not targeted at actual five-year-olds, we felt the nature of ELI5 could be a good fit for the BabyLM Challenge. We chose not to use the existing ELI5 dataset (Fan et al., 2019) as it is focused more on question answering and we needed text for pretraining that met our dataset objectives.

We obtained all of the posts to the ELI5 subreddit from June 2005 to December 2022 from The-Eye.eu Reddit archive[3]. We filtered this set of posts to leave only top-level comments, which are direct replies to questions, by searching for posts where the link ID matched the parent ID in the metadata. Only top-level comments are required by the subreddit rules to be simplified explanations. We then sorted the remaining posts in descending order based on the score they obtained through the built-in user voting system on the subreddit. In the event two posts had the same score, the more recent post came first in the sorted list. We applied a basic filter that removed posts containing any of the profane words in a 28-word list. We also removed posts that included "https" to filter out hyperlinks from our training data. This profanity filter is fairly simple and it is not likely that it removed all instances of profanity within the dataset (e.g. alternatively spelled profanity). However, profanity is a part of language and even children are exposed to a non-zero amount of profane language during their developmental years. Thus, we did not conduct further filtering beyond the simple list.

We created training sets of 10M and 100M

[2] www.reddit.com/r/explainlikeimfive

[3] https://the-eye.eu/redarcs/

words. We computed the number of words in each post by splitting the text on any white space character and summing the number of text segments that contained any alphanumeric character. We developed this method to count towards the limit any word that contains meaning. It does not count punctuation that stands alone, such as dashes. Working down the sorted list of posts, we added posts to each training set as long as it did not cause the sum to exceed the total word capacity. Due to this, our 10M word training set is a proper subset of the 100M word training set.

3.2 Curriculum Development

Classic masked language model training involves randomly masking tokens from the training data that is fed to the model. With a limited amount of training data, we sought to develop a curriculum that more closely mimicked how a child might learn language. Using a timeline for normal child language development (LaGreca, n.d.; Roseberry-McKibbin and Hegde, 2006), we developed the following steps:

1. Interjections, nouns, and personal pronouns
2. Conjunctions
3. Subject-verb-object structures, first person singular pronouns, plurals, and simple verb forms
4. Adjectives, plural proper nouns, possessives, wh-determiners, and pronouns
5. Complex verbs and possessive endings
6. Adverbs, particles, and complex adjectives
7. All other parts of speech

The curriculum was cumulative, so each step in the training contained the additional parts of speech for that step and all previous steps.

We used the Natural Language Toolkit (NLTK) library (Hardeniya et al., 2016) to implement part-of-speech (POS) tagging on our datasets. We converted each sentence in our training data into individual words and obtained the POS tag for each word from the toolkit. We then created a custom function that selectively masked words based on their grammatical categories. For each training example, we drew masked words from the pool of tags for the current curriculum step until either 15% of the total words were masked or all candidate words were used.

Table 1: Hyperparameters used for training

	100M	10M orig.	10M redo
Learn. rate	5e-5	5e-5	1e-4
Optimizer	AdamW	AdamW	AdamW
LR Profile	Linear	Linear	Cosine
Warmup	n/a	n/a	500

3.3 Base model selection and computing parameters

We used a BERT (Devlin, 2018) model with 6 layers, a hidden dimension of 768, and 12 heads to create a 51.2M parameter model. Following RoBERTa (Liu, 2019) and GPT-2 (Radford et al., 2019), we used a *byte pair encoding* (BPE) tokenizer to tokenize the inputs. We built tokenizers with 10K and 50K vocabularies for each of the 10M and 100M corpora, respectively. We used two sets of hyperparameters: the original set used for initial model training and an updated set that we used for a final training pass. These hyperparameters are compiled in Table 1. These parameters were largely arbitrarily chosen based on past experience by the authors and we note that there are probably additional changes to these design choices that could be made to improve performance.

We used a variety of GPUs and workstations to train and evaluate our models, including six 40Gb A100s, an A6000, two RTX2080Tis, and two RTX3090 GPUs. We estimate our combined GPU-days at around 30 days. Due to the varying VRAM available on each of these GPUs, batch sizes were not consistent between the training of different models and we note this as a weakness in our study. Past experience from one of the authors has shown that batch size is a particularly important parameter for small datasets as a bigger batch size smooths the loss landscape and reduces the capacity of the model to learn from individual examples. Private conversations with some industry members have suggested that in very small datasets, it's sometimes desirable to fine-tune with a batch size of one in order to learn the distribution of the data. However, due to the time constraints of this challenge, we maximized batch size to make use of the available GPUs and did not well-control or study it.

4 Results

In this section, we share the results of our models on the BabyLM shared task. We have attempted

Table 2: **Evaluation scores** The overall evaluation metrics we computed for all trained models. We did not train or evaluate models with a '_provided' suffix and the results presented come from the challenge organizers. The first BERT_10M_eli5_curr run results files do not have available GLUE scores and therefore no macroaverage.

Model	BLiMP	BLiMP Supplement	EWoK	GLUE	Macroaverage
BERT_10m_base	54.6	56.5	47.3	66.1	56.1
BERT_10m_eli5	54.7	56.5	49.9	66.8	57.0
BERT_10m_eli5_curr_mask_redo	55.6	56.1	**50.8**	**67.3**	57.5
BERT_10m_eli5_curr_mask_orig	51.25	52.23	48.0	xx.x	xx.x
LTG-BERT_10M_provided	**60.6**	**60.8**	48.9	60.3	**57.7**
BERT_100m_eli5	55.4	54.0	51.5	66.7	56.9
BERT_100m_eli5_curr_mask	60.2	56.8	**53.0**	67.7	59.4
LTG-BERT_100M_provided	**69.2**	**66.5**	51.2	**68.4**	**63.8**

to disentangle the impacts of the two approaches combined, although due to training time we were not able to do a full ablation study. First, we discuss the impact of the newly scraped dataset. Second, we share the results of the curriculum masking approach and discuss why it appears to outperform the typical MLM pre-training approach.

4.1 ELI5 dataset

Findings by Meta (Xie et al., 2024) show that having a high fraction of internet scraped data is generally the key to the highest performing language models. We decided that we would try to go with a solely internet-based training dataset in an attempt to take advantage of this effect. Following anecdotes from the training of the original Stable LM (Bellagente et al., 2024), which used a high fraction of Reddit-based training data and had poor performance, our data cleaning removed usernames (suspected to be responsible for strange tokenizer performance in Stable LM and GPT-2). To be able to identify the impact of our ELI5 dataset, we trained an identical model on the baseline dataset provided by the BabyLM organizers. In Table 2, the *BERT_10m_base* and *BERT_10m_eli5* entries show the baseline and ELI5 data evaluation results, respectively. LTG-BERT scores provided by the organizers were included as a fair comparison for the encoder-only BERT model we used. BLiMP scores show barely any difference, suggesting that grammatical phenomena are represented similarly in both datasets. EWoK, a benchmark evaluating world knowledge, shows improved results with the ELI5 dataset, which the authors find to be a reasonable outcome due to the simplistic explanations that capture world knowledge found in many ELI5 responses. SuperGLUE evaluations also show modest improvements from use of the ELI5 dataset, potentially indicating that the ELI5 data teachesa better language understanding than the baseline training dataset. Comparing both to the LTG-BERT results provided by the competition organizers (*LTG-BERT_10M_provided*), the BLiMP results of both BERT models are lower, but all other metrics are higher for our models.

When looking at the 100M models, they appear to underperform the provided model's results on BLiMP, be somewhat comparable on GLUE, and greatly outperform the provided model on EWoK. We did not spend as much time adjusting hyperparameters for or rerunning the 100M data, so it is likely there is a lot of room for improvement. However, despite this, these results reinforce our finding that ELI5 explanations help teach world knowledge to a language model.

4.2 Curriculum Masking

As described in Section 3.2, the curriculum masking process gradually introduced the model to new parts of speech, while continuing to train on the previously introduced parts of speech. Training results from the improved training hyperparameters (*BERT_10m_eli5_curr_mask_redo* in Table 2) (the hyperparameters listed in Table 1) are shown in Figure 1. The learning rates follow a cosine decay with a warmup period that resets every time a new part of speech is introduced. Loss shows small increases with the introduction of new parts of speech and then gradually decays as is typically expected. The gradient norm increases as the parts of speech become more complex, indicating that the model is not learning as well with

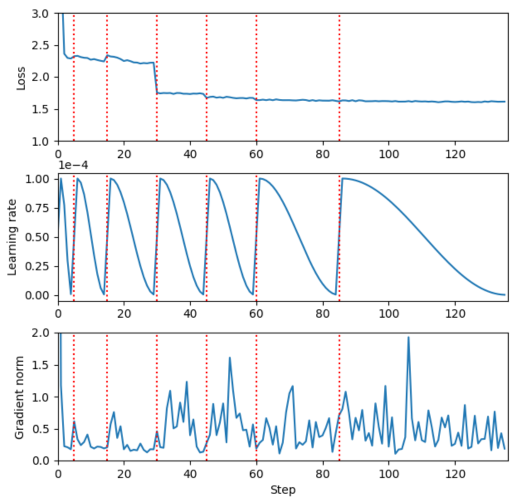


Figure 1: Curriculum masking training performance for 10M ELI5 training.

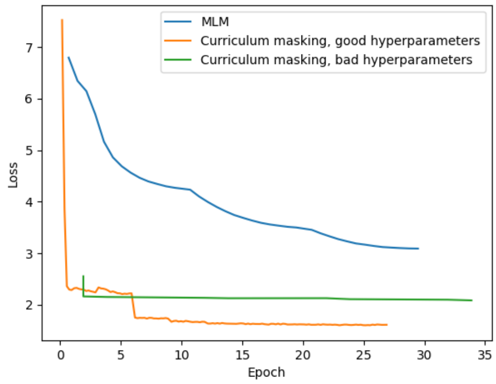


Figure 2: Loss curve comparison for curriculum-based and traditional masked language modeling with 10M model.

a very limited set of parts of speech but during the intermediate and later stages of the curriculum it learns quite effectively. By comparing with two other loss curves in Figure 2, we demonstrate that this loss curve outperforms either a linearly decaying learning rate that resets with each new POS or a typical masked language modeling approach with linearly decaying learning rate with three restarts (*BERT_10m_eli5_curr_mask_redo* and *BERT_10m_base*, respectively, in Table 2). We note that this is not a strong comparison as the logging rates do not match due to the batch size mismatch, but the general trends may be helpful for the reader. Importantly, we note that by focusing the model on learning specific aspects of language first, we are able to accelerate the learning of the more complex language aspects introduced later.

When looking at evaluation scores in Table 2, with the better hyperparameters, we demonstrate that the combination of the ELI5 data with the curriculum masking provides the best performance overall of any 10M model we evaluated. We note that BLiMP performance was comparatively poor for all of the models we trained, relative to the provided scores of the baseline model. For the 100M models, the curriculum masking improved results beyond just using the ELI5 dataset, although with poor BLiMP performance, the improvements to EWoK weren't able to increase the average score above the baseline model results provided by organizers.

5 Conclusions

In our submission to the 2024 BabyLM challenge, we focus on using an internet-based training dataset that mimics language that would be directed at youth as well as utilizing a developmentally plausible pre-training approach that allows the model to learn specific parts of speech on a schedule. We show that by combining both of these approaches, we can outperform the baseline provided by organizers on the STRICT-SMALL track (10M word limit) of the challenge, although we did not succeed at outperforming the baseline for the SMALL track (100M word limit). Due to a lack of hyperparameter optimization, there is probably a lot of improvement that could be made using curriculum masking, especially considering different masking ratios or masking ratio schedules. One other possibility we are interested in is varying the POS acquisition order and experimenting with the use of training the model on a mix of the POS on the schedule as well as some other words. Testing the curriculum masking concept on an autoregressive model would be an obvious thing to try as well.

Our findings help reinforce the idea that using internet-scraped data provides highly useful data for teaching a language model language understanding as well as world knowledge. Additionally, our proposed method of curriculum masking introduces a new method of curriculum learning that shows accelerated learning in our tests on a small dataset.

Limitations

The biggest limitation of this work is that it largely relies on two sets of hyperparameters and does not thoroughly explore the hyperparameter space in order to determine how stable and useful our proposed training method is. Masking rates were not explored at all and there are most likely masking rates or schedules for them that would further improve model training and performance. We have tried to explore the impact of our data and training method separately by running a partial ablation study, but we did not consider the impacts of data on our hyperparameter selection. Batch size is often noted as a powerful "knob" for tuning performance and due to the mismatched GPUs used for different training runs, we did not control this well and therefore are not able to quantify its impact on our model performance. There is also a dependence on the performance of the POS tagger and we don't have a good assessment of the performance of the POS tagger used without having labeled data from our dataset.

Ethics Statement

The authors are not anticipating any major ethical concerns with publishing this work. We propose a slight modification to the widely-used MLM pretraining task as well as a version of a publicly available dataset. We note that our use of the ELI5 Reddit data encourages the continued use of scraped internet data to train language models, which has been noted to potentially lead to self-training on generated content as more internet content becomes generated by language models. The long term impacts of this are not fully understood yet, but it is likely that it may be somewhat detrimental to both future model performance and, thus, internet content.

Acknowledgments

This work was supported by the Michigan Tech Institute of Computing and Cybersystems and by grant funding received from NIH/NIDCD R01DC009834.

References

Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, et al. 2024. Stable lm 2 1.6 b technical report. *arXiv preprint arXiv:2402.17834*.

Bastian Bunzeck and Sina Zarrieß. 2023. Gpt-wee: How small can a small language model really get? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 35–46.

Justin DeBenedetto. 2023. Byte-ranked curriculum learning for babylm strict-small shared task 2023. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 198–206.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

Nitin Hardeniya, Jacob Perkins, Deepti Chopra, Nisheeth Joshi, and Iti Mathur. 2016. *Natural language processing: python and NLTK*. Packt Publishing Ltd.

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*.

Andrei Jarca, Florinel-Alin Croitoru, and Radu Tudor Ionescu. 2024. Cbm: Curriculum by masking. *arXiv preprint arXiv:2407.05193*.

Lauren LaGreca. n.d. Normal language development for young children. https://www.lispeech.com/normal-language-development-young-children/. Accessed: 2024-09-19.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Richard Diehl Martinez, Zebulon Goriely, Hope McGovern, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. Climb: Curriculum learning for infant-inspired model building. *arXiv preprint arXiv:2311.08886*.

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.

Celeste Roseberry-McKibbin and Mahabalagiri N Hegde. 2006. *An advanced review of speech-language pathology: preparation for PRAXIS and comprehensive examination*. ERIC.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal

Linzen, et al. 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*.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2024. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36.

A Dataset examples

A few semi-cherry picked examples are shown for some of the ELI5 data and some of the provided baseline training data. It can be seen that the internet-based text of ELI5 is more coherent and provides a better textual training example (in the subjective opinion of the authors) than the transcribed text that is formatted in a variety of ways. Whether or not an explanation given at a level appropriate for a five year old is equivalent to what a five year old actually experiences is debatable, but from the language modeling perspective it is likely that the transcribed text may cause the model to learn strange behaviors that are not reflective of actual language usage.

ELI5 Sample response 1

The joke answer is so that the water doesn't hit you square in the face.

The real answer is that shapes with sharp corners are structurally weak. Arcs and circles are very strong shapes. If port holes were squares, the openings would get damaged and worn out sooner.

ELI5 Sample response 2

Caffeine works in two ways to make you feel that way.

First it prevents the brain from telling you that you are tired. You can think of your brain as a bunch of locked boxes with different things inside of them. Some of these boxes have things that make you happy, others make you sad. Some have things that tell you it is time to go to sleep. Caffeine jams itself into the lock on the sleepy time box so that your brain can't open it. That keeps you from feeling tired.

Caffeine also can help open the box that tells your body to go into extra energy mode. Things like your heart can work faster or slower depending on what you need. If you are sitting on the couch watching TV it's going to go slower, if you are outside working it's going to speed up. Caffeine tricks the body into thinking it needs to go into extra energy mode. Caffine doesn't create this energy, the body is just using what it has stored more quickly. Not really any different from you step on the gas in a car. You are telling it to burn more fuel and go faster.

ELI5 Sample response 3

You know when you're going on vacation, and you're packing, but you still need to use some of the stuff you need to pack, so instead of putting it all into your suitcase, you set some of it next to your suitcase, or leave it out on the counter, so you don't forget it, but you can still use it without having to completely unpack it from your luggage?

That's sort of how a USB drive works. Sometimes you tell the computer to "pack" data onto the drive, and rather than put it all on there right away, it might end up caching some of it to be written later.

When you just rip out the drive, you risk pulling it before all of your data is "packed" onto the drive.

When you click "safely remove" it runs around the house and packs up all the stuff it left out, and gets it all into the luggage for you before you disconnect it.

BabyLM Provided 10M_Train Sample 1

Have you ever seen anybody completely obscured by her own smoke, it's Sharon.
.
Chuck us the water would you?
She's a bit of a goer as well int she?
Is she?
Isn't she?
Yeah but
Didn't she order a punch so she was drunk ?
No, that was Tracey.
I thought Tracey and Sharon used to get drunk at lunchtime on a Friday and have a punch up.
No.
Only Tracey would do that.
Our Trace.
Ah.
Oh dear.
Oh.

BabyLM Provided 10M_Train Sample 2

THIS IS EXACTLY WHAT I'M TALKING ABOUT.
I'M NOTHING BUT A BIG MAC IN A BATH TOWEL.
JOEY, I'M NOT A HAMBURGER.
I HAPPEN TO BE A HUMAN BEING.
JESS, BUDDY, AS LONG AS I'M THE DIRECTOR,
YOU WILL BE TREATED WITH DIGNITY AND RESPECT.
THANK YOU.
OK, HOSE HIM DOWN.

BabyLM Provided 10M_Train Sample 3

*CHI: Eve tapioca hot.
MOT: uhhuh.
CHI: hot.
MOT: mhm.
CHI: and cool.
MOT: and cool yes.
MOT: by the time you have lunch it'll be cool.
CHI: that?
MOT: what is that?
MOT: vanilla.
CHI: vanilla.
MOT: vanilla.
CHI: vanilla.
MOT: vanilla.
CHI: Eve play bouillon cube.

WhatIf: Leveraging Word Vectors for Small-Scale Data Augmentation

Alex Lyman and **Bryce Hepner**
Brigham Young University
alexlyman@byu.edu

Abstract

We introduce **WhatIf**, a lightly supervised data augmentation technique that leverages word vectors to enhance training data for small-scale language models. Inspired by reading prediction strategies used in education, **WhatIf** creates new samples by substituting semantically similar words in the training data. We evaluate **WhatIf** on multiple datasets, demonstrating small but consistent improvements in downstream evaluation compared to baseline models. Finally, we compare **WhatIf** to other small-scale data augmentation techniques and find that it provides comparable quantitative results at a potential tradeoff to qualitative evaluation.

1 Introduction

The use of Large Language Models (LLMs) has exploded in the recent past, with LLMs becoming the state of the art for most NLP tasks. While statistical models of language have been around for decades (Markov, 2006), the introduction of the Transformer (Vaswani, 2017) set the stage for a new era in language modeling.

Early Transformer-based language models such as BERT (Devlin, 2018) and GPT (Radford, 2018) are very small by today's standards, with a few hundred million parameters each. In the intervening years, models have grown exponentially both in number of parameters and number of training tokens. These increases in size have been accompanied by increases in performance, with abilities emerging as a consequence of model scale (Wei et al., 2022). Current state of the art models tend to have tens of billions to hundreds of billions of parameters and are trained on trillions of training tokens.

In this paradigm of increasing scale, there has been relatively little focus on small-scale language modeling, which tends to be restricted to domains such as low-resource machine translation.

The BabyLM challenge (Choshen et al., 2024) seeks to focus researchers on very small-scale language modeling. The challenge involves using either a 10 or 100 million word "developmentally plausible" corpus (Warstadt et al., 2023), with 100 million words being roughly amount of words a child hears before reaching adulthood. Working at this small scale enables researchers to focus on cognitively inspired methods of language modeling as well as to iterate on language modeling experiments, which is impractical at 100-billion parameter scales.

While much recent focus on language modeling has involved scaling up parameter and training token counts, these approaches have drawbacks, including environmental concerns and inaccessibility of hardware (Bender et al., 2021). As a consequence, there has been a recent focus on mid-scale language modeling, creating models that can be run locally on devices such as consumer PCs or smartphones. This research has been promising. Microsoft's Phi models (Li et al., 2023; Abdin et al., 2024) boast impressive performance on many language modeling benchmarks, in spite of having only a few billion parameters. Phi's major innovation is using only "textbook quality data", curated from only high-quality sources rather than semi-filtered data of dubious quality scraped from the internet.

At a much smaller parameter scale, Eldan and Li (2023) trained very small transformers on TinyStories, a synthetic dataset of children's stories. In spite of parameter counts below 10 million, these tiny models were able to generate coherent text with real world knowledge and logic.

The trend towards improving data quality and quantity rather than solely scaling model parameters has also been applied successfully to larger-scale language modeling. Llama 3 (Team, 2024) attributes its significant improvements in performance over Llama 2 (Touvron et al., 2023) not to

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 229–236

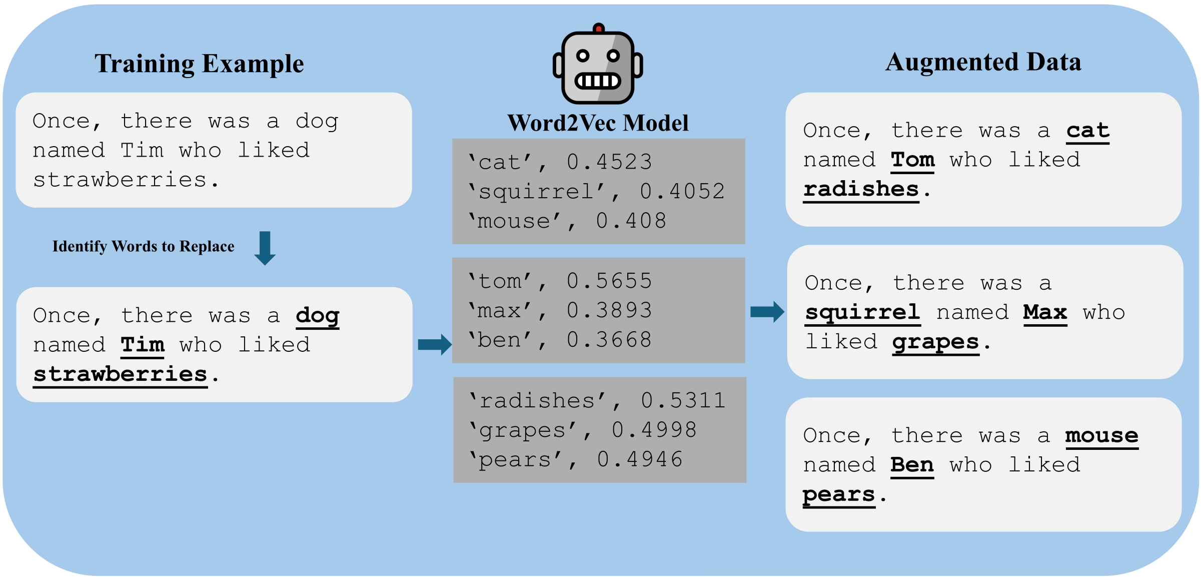


Figure 1: Illustration of the data augmentation technique.

changes in architecture, but to "improvements in data quality and diversity as well as by increased training scale." Using higher quality input, Llama 3 was trained on roughly ten times as many tokens as Llama 2.

Our research is motivated by both the language modeling research on training data, as well as children's processes of language acquisition.

Training on more data has a human analog. One of the strongest predictors of children's linguistic development is the amount and type of language they hear (Weisleder and Fernald, 2013). Children who hear more words tend to have larger vocabularies, which correlates with better educational outcomes later in life (Hart et al., 1997; Hoff, 2003).

We introduce **WhatIf**, a lightly supervised data augmentation technique that uses word vectors to augment training data. **WhatIf** substitutes words in the training corpus for semantically similar words, enabling our baby models to consider novel yet similar text to the training data.

WhatIf is inspired by a method of improving reading comprehension called predicting. With this strategy, teachers instruct students to periodically ask questions about the text. These questions can be predictions about what might occur later in the text or counterfactuals, which usually take the form of "What If?" questions. Teaching with prediction strategies improves reading instruction outcomes. Both children (Küçükoğlu, 2013) and second language learners (Ali and Razali, 2019) show improved reading comprehension when employing prediction strategies.

Our method is lightly supervised, and requires a small part-of-speech dictionary, which we count as part of our token budget. This too has an analog in real-world language acquisition. Children receive explicit grammatical knowledge. For example, children who produce ungrammatical speech are often corrected by a parent or caregiver. Children who receive explicit grammatical instruction and have explicit grammatical awareness tend to develop better linguistic skills (Ehri et al., 2001).

We perform experiments and show that **WhatIf** increases model performance on a variety of evaluation tasks, and performs comparably to other small-scale Language Model data augmentation techniques, though these quantitative gains come at a cost to text quality.

2 Methods

2.1 Data Augmentation

The core of the data augmentation technique is word vectors. We use the Word2Vec algorithm (Mikolov et al., 2013) to create semantic embeddings for each word in our training corpus. When trained over a sufficiently large corpus, Word2Vec embeddings cause similar words to end up with similar vector representations in the high-dimensional space. If the words are sufficiently semantically similar, changing one word for its nearest neighbor should preserve most of the sense and meaning of the text, while still creating novel, useful training examples.

We first split the corpus into sentences, then use the sentences to train a Word2Vec model. Then for each training example, we select p percent of the content words at random, excluding function words, which do not have grammatical equivalents. For each of the chosen words, we use the word vector model to select the nearest neighbor via cosine similarity, which is most semantically similar. Then, we check whether both the word and its candidate replacement have the same part of speech. If so, we replace each occurrence of the word in the training example with the candidate replacement. Otherwise, we repeat with the next nearest neighbor until we select a viable candidate or reach a preset distance threshold in vector space. The use of a threshold prevents the selection of semantically distinct words that happen to be nth near neighbors.

Once a viable replacement is selected, each occurrence of the word in the training example is replaced. This guarantees semantic continuity throughout the training example. This process can be repeated any number of times to increase the amount of data available to the model, starting from the gold standard each time. At each iteration the word vector model selects less similar words, theoretically enabling us to create large amounts of data of decreasing quality.

This technique does not guarantee grammatical or correct training examples. For example, according to the word vector model trained on our TinyStories corpus, the most similar word to *old* is *elderly*.

This works in contexts like:

*The **old** man / the **elderly** man.* ✓

However, this causes problems in contexts such as:

*The **old** castle / the **elderly** castle.* X

2.2 Model and Training Details

Because our primary focus is on training data, we select a simple model for our experiments based on the GPT-2 architecture (Radford et al., 2019). Although previous competition results show that GPT-2 is not the best architecture for small-scale language modeling, we choose it for ease of use, familiarity, and reproducibility.

We use the same training setup for all models, a version of the GPT-2 Small checkpoint with reduced size. While GPT-2 Small has an inner dimension of 768, we halve that size for an inner dimension of 384. We also halve the context window size from 512 to 256. Our models each have 26,000,640 trainable parameters.

To be able to iterate over many experiments, our models are optimized to train quickly, with a potential tradeoff in absolute performance. This is achieved both by reducing the size of the models and training them in FP16 precision.

To maximize training speed, all models were trained across 8 NVIDIA A100 GPUs with a batch size of 16, with a torch manual seed set to the same value for each model.

Across all experiments, batches are shuffled between each epoch.

We train a tokenizer on each dataset using HuggingFace's BPE implementation [1], with a vocabulary size of 12000.

Because different data augmentation techniques result in training sets of different sizes, we checkpoint and evaluate using steps instead of epochs. We train all models for roughly the equivalent of 100 epochs of the un-augmented dataset, and evaluate every 5,000 steps. Model performance tends to peak between 10,000 and 25,000 steps. (20-50 non-augmented epochs) To maintain consistency, we evaluate the 25,000 step checkpoint of each model for final evaluation.

2.3 Dataset Details

To ensure our results are not dataset dependent, we perform all experiments on two datasets. The first is a lightly filtered version of the BabyLM 2024 Strict-Small dataset comprising roughly 9,300,000 tokens. This data includes transcribed speech, narrative, and instructional texts. The second is a subset of the TinyStories dataset with roughly 9,950,000 tokens. These stories are synthetic data generated by GPT-3.5 and GPT-4, and are all short narratives with a target audience of 3-year-old children.

For each dataset, we use a small portion of the token budget for a part-of-speech dictionary. TinyStories has 12,233 key-value pairs for 24,466 total tokens, and BabyLM has 128,124 key-value pairs for 256,248 tokens. The part-of-speech dictionaries use Penn Treebank P.O.S. tags (Marcus et al., 1993). We also use the 159-word list of English stopwords from the NLTK package (Bird et al., 2009). In both cases, the count of training words and data augmentation materials falls under the 10 million token budget.

[1] https://github.com/huggingface/tokenizers

Models are trained on the training samples, with the dictionaries being used only for the data augmentation process.

2.4 Data Pre-Processing

Natural language occurs in context. In initial experiments, we found that joining lines from the BabyLM dataset into chunks led to large gains over to passing training examples line-by-line. Using contextual chunks enables the model to learn features of natural language such as conversational turn-taking. Across subsets of the training data, lines vary wildly in size. For each subset of the corpus we join a different number of lines to create each training example, with the goal of creating chunks of around 150 words. The 150 word mark was chosen because it enables most tokenized examples to fit within the 256 token context window. It is also close to the average number of words in the stories from the TinyStories dataset, allowing for an apples-to-apples comparison between models trained on both datasets.

3 Results

We perform a variety of experiments to probe the efficacy of our data augmentation technique. All experiments are performed on both the TinyStories dataset and the BabyLM Strict-Small dataset.

For our baselines, we train models using a standard language modeling approach. These examples occasionally need to be truncated, but thanks to the data pre-processing, the overwhelming majority of samples do not require truncation.

Because each pass of the augmentation process results in lower quality data, we experiment with how many passes of augmented data we create, $n = 5$ or $n = 10$ passes. For every n passes of augmented data, we also include one pass of the non-augmented gold standard data. This means only $\frac{1}{6}$ or $\frac{1}{11}$ of data seen while training on augmented data is gold-standard data. We also experiment with the percent of content words to replace, either 50%, which leaves the sample recognizable, or 100%, which drastically changes the training example. Examples of different degrees of augmentation can be found in the appendix.

3.1 Quantitative Results

We evaluate our models using the competition's default evaluation harness (Gao et al., 2023) and metrics: BLIMP (Warstadt et al., 2020), EWOK (Ivanova et al., 2024), and GLUE.

As shown in Table 1, **WhatIf** provides a small but consistent gain of 1 to 2 percentage points over the baselines.

Interestingly, benchmarks offer no clear trend as to the ideal hyperparameters for the data augmentation technique. The 5 pass models usually outperform their 10 pass counterparts, but by such a small margin that no clear conclusion can be drawn. While all augmented models outperform the baseline, there is not a clear winner.

We compare our results with a variant of the Contextualizer (Xiao et al., 2023), one of the best-performing data augmentation methods from the 2023 challenge. In our variant, **Contextualizer-like**, before each training pass we tokenize the whole dataset and shuffle the training examples. We then concatenate the tokenized samples and break them into 256-token chunks. We find that **Contextualizer-like** performs at a similar rate to **WhatIf** with a 1-2 percentage point increase over the pad and truncate baseline.

Finally, we ensemble our data augmentation technique with the **Contextualizer-like** algorithm to see if combining the methods causes an additional gain in performance. The results show that while both **WhatIf** and **Contextualizer-like** provide gains in performance, ensembling the two of them does not provide additional benefit.

3.2 Qualitative Results

Although **WhatIf** produces a small quantitative improvement as measured by benchmarks, models trained on augmented data can produce qualitatively worse text. To demonstrate this, we generate short completions to the prompt `Once upon a time` with three of our models trained on the TinyStories dataset with varying degrees of augmentation: the baseline model, the 5-pass-replace-50% model, and the 10-pass-replace-100% model. Samples are generated using top-k sampling with a temperature of 1 and a k of 20.

Baseline Model

```
Once upon a time, in a big forest,
there was a little bird.  The little
bird lived in a cage.  The bird had
a mommy bird.  The mommy bird could
not see the little bird in the cage.
The mommy bird was sad...
```

The baseline model generates a reasonable narrative, comparable with the output from the original TinyStories paper.

		BLIMP	BLIMP Sup.	EWOK	GLUE	Average
TinyStories	Baseline	55.9	52.9	51.3	59.6	54.925
	5 Pass Replace 50%	58.8	57.1	50.6	59.8	**56.575**
	10 Pass Replace 50%	58.8	54.2	50.9	60.6	56.125
	5 Pass Replace 100%	59.2	54.9	50	61.4	56.375
	10 Pass Replace 100%	58.3	54.1	50.9	60.7	56
	Contextualizer-like	59.1	54	51.5	60.1	56.175
BabyLM	Baseline	63.7	54.6	49.7	60.5	57.125
	5 Pass Replace 50%	64.5	56.6	50.8	60.9	58.2
	10 Pass Replace 50%	63.9	56.5	50.6	60	57.75
	5 Pass Replace 100%	66.3	59	50.6	60.4	**59.075**
	10 Pass Replace 100%	64	56.1	51.2	60.8	58.025
	Contextualizer-like	66.9	56.4	51.7	60.7	58.925

Table 1: Results of baseline and augmented models, evaluated at the 25,000 step checkpoint.

10-pass-replace-100%

```
Once upon a time, there was a child
named True.  True started to travel
with his brother, Bob.  They were
very stupid at riding games.  One
day, True returned hurt while they
worked.  Bob felt confused.  He said
to True, "I am sorry, let's travel
to my parent....
```

This text is lower in quality. We see examples of grammatical constructions that make no semantic sense, "`very stupid at riding games`" as well as poor world knowledge, e.g. `True` is not a normal name.

5-pass-replace-50%

```
Once upon a time, there was a little
kitten named Amy.  Amy liked to cook
with her mom.  One day, they decided
to cook a big salad for lunch.  Amy
was very happy.  Amy's mom told her,
"Amy, can you put the salad in the
oven?" Amy opened the oven and put
the salad in the oven...
```

While the overall story lacks some world knowledge (salad is not typically cooked in an oven), this output suggests this somewhat augmented training mix may be a reasonable compromise between quantity and quality, though further experiments are necessary to identify the ideal training mixture.

4 Discussion

Just like children, small-scale language models benefit from additional data. **WhatIf** shows mild but consistent benchmark improvement above the baseline across datasets.

We expect this to improves performance on benchmarks by exposing the LM to new scenarios during training. In practice, the augmented data is sometimes fairly low-quality. As a consequence, the LM can learn incorrect facts about the world. For example, augmentation may replace the word *Mom* with *Dad*, without replacing gendered pronouns *she* with *he*. This does not seem to have a large negative effect on the model's grammatical abilities, since BLIMP and BLIMP supplement scores improve with **WhatIf** augmentation. However, EWOK scores do not improve decreasing slightly when applied to the TinyStories dataset. We suspect that the LM earns incorrect information about the world from bad correlations in the augmented data.

The fact that both **WhatIf** and **Contextualizer-like** provide similar gains suggests that manipulating the training data in some well-informed way provides modest performance gains. Since we observe diminishing returns when ensembling both methods, this might mean that both methods are acting on a similar axis to make the training data more useful to the model.

5 Limitations and Future Work

This work is only a partial realization of the underlying idea that data augmentation with word vectors could improve model performance. We suspect that small changes to the data augmentation algorithm could bear significant fruit. An additional round of validation to improve the coherence of the augmented data would probably help.

Our analysis is limited to autoregressive language models, and experiments should be repeated with masked language models. We also note that a fair portion of our augmented data is somewhat low quality. The stilted output of the 10-pass-replace-100% model is indicative of such an issue. Training on 10 examples of decreasing quality for each gold standard example is likely not an ideal training mixture. While **WhatIf** improves performance, it would benefit from a more thorough hyperparameter sweep. Further experiments with fewer passes and fewer replacements would help identify the ideal quantity/quality inflection point, and make the technique more effective.

Acknowledgments

We gratefully acknowledge the support of the National Science Foundation (Grant 2214708).

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Aziza M Ali and Abu Bakar Razali. 2019. A review of studies on cognitive and metacognitive reading strategies in teaching reading comprehension for esl/efl learners. *English Language Teaching*, 12(6):94–111.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

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*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Linnea C. Ehri, Simone R. Nunes, Dale M. Willows, Barbara Valeska Schuster, Zohreh Yaghoub-Zadeh, and Timothy Shanahan. 2001. Phonemic awareness instruction helps children learn to read: Evidence from the national reading panel's meta-analysis. *Reading Research Quarterly*, 36(3):250–287.

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

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Betty Hart, Todd R Risley, and John R Kirby. 1997. Meaningful differences in the everyday experience of young american children. *Canadian Journal of Education*, 22(3):323.

Erika Hoff. 2003. The specificity of environmental influence: Socioeconomic status affects early vocabulary development via maternal speech. *Child development*, 74(5):1368–1378.

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 Akyürek, 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. *Preprint*, arXiv:2405.09605.

Hülya Küçükoğlu. 2013. Improving reading skills through effective reading strategies. *Procedia - Social and Behavioral Sciences*, 70:709–714. Akdeniz Language Studies Conference, May, 2012, Turkey.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

Andreĭ Andreevich Markov. 2006. An example of statistical investigation of the text eugene onegin concerning the connection of samples in chains. *Science in Context*, 19(4):591–600.

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

Alec Radford. 2018. Improving language understanding by generative pre-training.

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.

Llama Team. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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 Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023. Call for papers–the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2301.11796*.

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.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Adriana Weisleder and Anne Fernald. 2013. Talking to children matters: Early language experience strengthens processing and builds vocabulary. *Psychological science*, 24(11):2143–2152.

Chenghao Xiao, G Thomas Hudson, and Noura Al Moubayed. 2023. Towards more human-like language models based on contextualizer pretraining strategy. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 317–326, Singapore. Association for Computational Linguistics.

A Appendix 1: Augmented Training Examples

Example augmented story, tenth pass, replace 100%

```
  Once upon a time, there was a horse named Tom.  Tom loved to speak with his pipe
and bite his package.  One day, Tom was playing in the restaurant with his best
friend, a little rabbit named Sam.  Sam swung the pipe and Tom hurried to pick it.
But this time, something unexpected happened.  Tom saw a great big rabbit.  The
rabbit lifted the package from Tom's stomach and rolled away.  Tom was uncomfortable.
Sam had an idea to supply Tom delighted again.  He lifted a big amount of sticker
and drew a lot of the great big rabbit with the package.  Tom loved the new lot and
started to bite it.  Now, Tom had a new rock to bite and speak with.  And they all
lived happily ever after.
```

Example augmented story, third pass, replace 50%

```
  One day, a boy named Tim discovered an yellow hoop.  He picked it up and met that
it was very pretty.  Tim wanted to play with the hoop, so he called his sister,
Sam.  Sam walked over, and they began to play a tag.  "Let's shoot the hoop into the
tube," asked Tim.  Sam agreed, and they grabbed turns shooting the hoop.  They were
having a picture of fun.  Suddenly, the yellow hoop stepped stuck in a tree.  They
tried to get it down, but it was too high.  Just then, a lamb named Lily walked by
with a big dictionary.  "What's that?" told Tim.  "It's a novel," asked Lily.  She
met the yellow hoop in the tree and had an idea.  She rolled the novel at the hoop,
and it jumped down.  Tim, Sam, and Lily were all surprised that the novel used get
the hoop down.  They all danced and played together for the rest of the day.
```

B Appendix 2: Evaluation Results for All Checkpoints

			BLIMP	BLIMP Sup.	EWOK	GLUE	Average
WhatIf	TinyStories	Baseline	55.9	52.9	51.3	59.6	54.925
		Aug 5 Pass Replace 50%	58.8	57.1	50.6	59.8	**56.575**
		Aug 10 Pass Replace 50%	58.8	54.2	50.9	60.6	56.125
		Aug 5 Pass Replace 100%	59.2	54.9	50	61.4	56.375
		Aug 10 Pass Replace 100%	58.3	54.1	50.9	60.7	56
	BabyLM	Baseline	63.7	54.6	49.7	60.5	57.125
		Aug 5 Pass Replace 50%	64.5	56.6	50.8	60.9	58.2
		Aug 10 Pass Replace 50%	63.9	56.5	50.6	60	57.75
		Aug 5 Pass Replace 100%	66.3	59	50.6	60.4	**59.075**
		Aug 10 Pass Replace 100%	64	56.1	51.2	60.8	58.025
Contextualizer-like **+WhatIf**	TinyStories	Contextualizer-like	59.1	54	51.5	60.1	56.175
		Aug 5 Pass Replace 50%	61.2	53.9	51.6	59.4	56.525
		Aug 10 Pass Replace 50%	58.6	53.4	50.8	59.5	55.525
		Aug 5 Pass Replace 100%	60.2	52	50.5	59.8	55.675
		Aug 10 Pass Replace 100%	61.8	54.1	50.9	60.8	**56.9**
	BabyLM	Contextualizer-like	66.9	56.4	51.7	60.7	**58.925**
		Aug 5 Pass Replace 50%	66.3	57.3	51	59.1	58.425
		Aug 10 Pass Replace 50%	66.2	58.4	50.9	60.1	58.9
		Aug 5 Pass Replace 100%	66.6	58.5	50.1	59.8	58.75
		Aug 10 Pass Replace 100%	64.9	58.4	50.9	59.5	58.425

Table 2: Results of all 20 models, evaluated at the 25,000 step checkpoint.

A surprisal oracle for when every layer counts

Xudong Hong[‡], Sharid Loáiciga[†] and Asad Sayeed[†]
[‡]Dept. of Language Science and Technology and Dept. of Computer Science, Saarland University
[†]Dept. of Philosophy, Linguistics, and Theory of Science, University of Gothenburg
`{xhong}@lst.uni-saarland.de`, `{sharid.loaiciga, asad.sayeed}@gu.se`

Abstract

Active Curriculum Language Modeling (ACLM; Hong et al., 2023) is a learner-directed approach to training a language model. We proposed the original version of this process in our submission to the BabyLM 2023 task, and now we propose an updated ACLM process for the BabyLM 2024 task. ACLM involves an iteratively- and dynamically-constructed curriculum informed over the training process by a model of uncertainty; other training items that are similarly uncertain to a least certain candidate item are prioritized. Our new process improves the similarity model so that it is more dynamic, and we run ACLM over the most successful model from the BabyLM 2023 task: ELC-BERT (Charpentier and Samuel, 2023). We find that while our models underperform on fine-grained grammatical inferences, they outperform the BabyLM 2024 official baselines on common-sense and world-knowledge tasks. We make our code available at `https://github.com/asayeed/ActiveBaby`.

1 Introduction

In this work, we describe our contribution to the "strict-small" task of the BabyLM Challenge of 2024 (Choshen et al., 2024) which follows up our contribution to BabyLM 2023 (Hong et al., 2023). Our effort this year focused on two activities: (1) testing the most successful contribution to BabyLM 2023, Every Layer Counts BERT (ELC-BERT; Charpentier and Samuel, 2023) under additional conditions and (2) implementing our training protocol, which we called Active Curriculum Language Modeling (ACLM) over our attempt at replicating ELC-BERT. We test ELC-BERT under more constrained conditions and explore whether the result is stable under other hyperparameter settings. Under very similar settings, we test our ACLM approach to see whether it exceeds the performance of our baselines on the BabyLM evaluation tasks.

Our intuition is that a human learner is an active participant in the environment of language acquisition (Fazekas et al., 2020; Masek et al., 2021). That children are not passive participants in L1 acquistion goes essentially without saying in contemporary developmental psycholinguistics—and with anyone who has interacted with a small child for any length of time—but the best artificial learners under data-constrained conditions (such as ELC-BERT) are trained in an entirely passive way. Their higher performance stems entirely from technical adjustments to the "training math". While this results in impressive performance, the insights it can give to the "whole picture" of how children acquire language from small data is limited, given what we know already about human development.

Instead, our over-arching hypothesis is that for every successful "passive" language modeling training algorithm, there is a way of scheduling the learning process that is more cognitively plausible or better-performing or both. This is not straightforwardly "classical" curriculum learning with the curriculum calculated or set in advance. Rather, it takes its inspiration from active learning (Zhang et al., 2022), where the learner (usually in a classification task) assesses its uncertainty on hitherto unseen items, and then asks for a human label, in a process that reduces the burden of labelling more training data than there are resources to label.

ACLM instead uses a cycle in which the learner trains an initial model from a small subset of the examples, and then iteratively adds to its dataset by using an uncertainty criterion over the items automatically, essentially creating a "dynamic" curriculum during the learning process (Bengio et al., 2009; Jafarpour et al., 2021).

The outcome of the overall BabyLM 2023 task participation (Warstadt et al., 2023) suggested that curriculum learning was not fruitful in exceeding the original baselines or in overall competitiveness on the BabyLM task as compared to model archi-

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 237–243
November 15-16, 2024

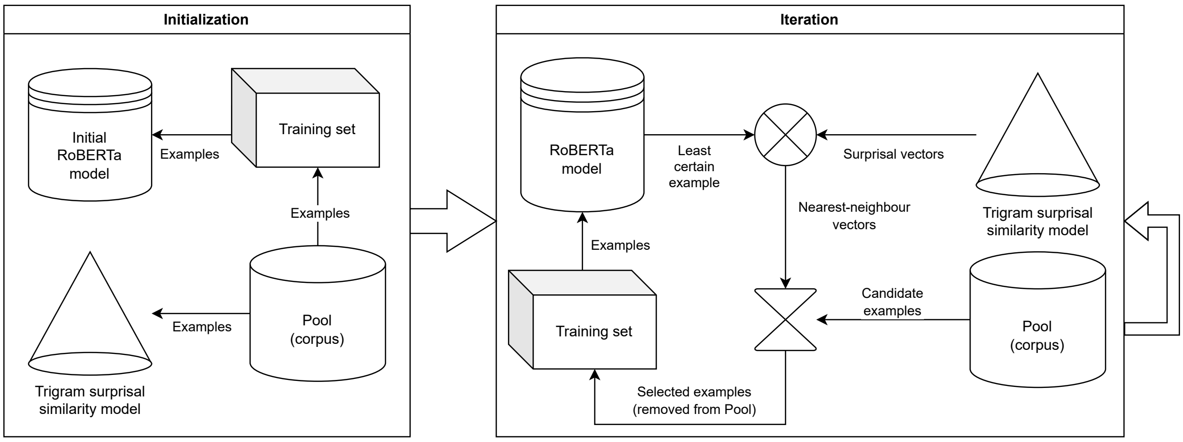


Figure 1: The architecture of our ACLM method from last year's submission, described in Hong et al. (2023). For this study, we modify the trigram surpisal similarity model to simply use the average sentence surprisal of the model under training, which is now ELC-BERT rather than RoBERTa.

tecture "tweaks" such as ELC-BERT. The present study suggests a more mixed picture: that the advantages from architectural modifications are highly sensitive to peturbations from hyperparameters, while a dynamically updated curriculum such as ACLM still may have the potential to augment a high-performing model architecture while retaining some connection to the interactive nature of human language acquisition.

2 Background

The BabyLM task necessarily involves the exploration of a very large solution space. In the previous year's challenge, we proposed an initial system, which we depict in fig. 1. Because we had to start somewhere, this involved design decisions based on educated guesses as well as a focus on efficiency and "getting it off the ground". The result of that effort was that our system ended in the "middle of the pack" and behind a baseline BERT model in the actual competition, but nevertheless resulted in insights that led us to consider how we can continue to explore this part of the BabyLM solution space, considering our expectations from a cognitive perspective.

Our previous system started with a blank RoBERTa model (Liu et al., 2019; Zhuang et al., 2021) which was initialized by training on a small subset of the training corpus. The remainder of the training corpus (the "pool") was processed via a trigram model into sequences of per-token trigram surprisal values, where suprisal is defined as the negative log-probability of the trigram ending in the given token. These sequences were resampled into an arbitrarily-chosen seven dimensions (which we later found was the average token length of the samples in the corpus), which we call the "surprisal signature".

At each iteration of training (several epochs), the RoBERTa model is queried about every sentence in the current training set, starting from the initial subset: which previously-seen utterance had the highest average surprisal. Then a k-nearest-neighbours process was used to sample the most similar surprisal signatures to the signature of the least certain sentence. These are added to the active training set, and the next iteration commences.

This process is different from the active curriculum learning process of Jafarpour et al. (2021). Jafarpour et al. develop a way to combine a human expert-designed curriculum with an active-learning informativeness criterion in order to select instances for humans to label. Since our training task is a language modeling task rather than a labeling or classification task, we can eliminate the human in the loop (effectively, the label is a word we already have in the text) and use the informativeness criterion to structure an automatic curriculum.

One obvious weakness of this process is that the surprisal space is static and does not reflect changes in the learner's estimation of what is surprising with respect to what is being learned. There are other weaknesses of this process, such as the seeming arbitrariness of the seven-dimensional vectors or even the use of surprisal as the criterion itself. However, in this update of our previous work, besides replacing RoBERTa with ELC-BERT, we focus

Algorithm 1 Initialization phase of this year's ACLM process.

```
Model ← new(ELC-BERT)
ActiveSet ← select_random(Pool, n_initial)
train(Model, ActiveSet, n_epochs)
SurprisalSet ← []
for all instances i in Pool do
    surprisals ← Model.surprisals(i)
    SurprisalSet.append(surprisals)
end for
```

Algorithm 2 Iterations of the ACLM process. The kNN procedure also removes the instances from the Pool.

```
for iter ← 0 to n_iterations do
    max_surprised ← TrainingSet[0]
    for all instances i in TrainingSet do
        orig_surprisal ←
            Model.surprisals(max_surprised)
        new_surprisal ← Model.surprisals(i)
        if orig_surprisal < new_surprisal then
            max_surprised ← i
        end if
    end for
    ActiveSet.update(SurprisalSet.kNN(
            max_surprised, k, Pool))
    train(Model, ActiveSet, n_epochs)
    SurprisalSet ← []
    for all instances i in Pool do
        surprisals ← Model.surprisals(i)
        SurprisalSet.append(surprisals)
    end for
end for
```

on the first weakness and implement a dynamic process as described below.

3 Learner-directed Active Curriculum Language Modeling

In the original formulation of ACLM, the "surprisal space" (the collection of "surprisal signature" vectors), was static throughout training, leading to a curriculum directed entirely by the model's uncertainty over each active learning iteration. From a cognitive perspective, this would be equivalent to a human learner whose expectations about the most educational thing in the environment never change from birth. We now propose an update to ACLM: we generate a new suprisal space at every iteration, which more closely matches the idea that a learner changes its view of the learning environment as it learns. In practical terms, this means that our ACLM model now re-evaluates the suprisal space using the ELC-BERT model itself, producing a new surprisal space reflecting the model's current "knowledge state". We view this as increasing the cognitive realism of ACLM and making it reflect a more learner-directed approach to acquisition.

We describe the ACLM procedure in this year's submission at a high level. We refer to "iterations" of the training procedure to be periods between updates of the active training set from the pool (corpus of as-yet-unseen training items). Multiple training epochs can take place during an iteration, meaning that the model may see the same training set without an update multiple times. Algorithms 1 and 2 provide an overview of the process, with the former describing the initialization process and the latter the iterative curriculum adaptation. The biggest procedural difference between this and Hong et al. (2023) is the use of the model itself to update the surprisals at every iteration.

In our previous submission, we split the corpus by utterance. This year, we follow the practice of the ELC-BERT implementation of having a sequence length of 128 tokens regardless of utterance boundaries. In the surprisal space, our dimensionality reduction proceeds to 7 dimensions (D7, as in our previous submission), 32 dimensions (D32), 64 dimensions (D64), and 128 dimensions (D128, essentially with no reduction). The reduction of the surprisal space no longer represents an attempt to equalize sentences of varying lengths through image resampling[1], since everything starts from 128 tokens.

4 Analysis

We list our results in table 1. The LTG-BERT baseline for the strict-small BabyLM 2024 task was trained with a batch size of 32786 and 8196 as well as corresponding sequence lengths of 512 and 128 (Samuel et al., 2023). The equivalent ELC-BERT run for BabyLM 2023 was also trained with a batch size of 8096 and a sequence length of 128 (Charpentier and Samuel, 2023). This is very resource-intensive, so we instead trained non-ACLM models with batch sizes of 32 and 512 (ELC-BERT B32 and ELC-BERT B512). Gradient accumulation was used as well to mitigate the smaller batch sizes (complete list of hyperparameters in Table 2, sec-

[1]We simply use the `resize` method from scikit-image.

tion A).

The original ELC-BERT still vastly outperforms both the BabyLM 2024 strict-small baselines as well as all of our models on BLiMP and GLUE. We will not attempt to explain ourselves why the 2024 baselines underperform the 2023 ELC-BERT submission[2] and focus our discussion on this year's baselines.

On BLiMP (Warstadt et al., 2020), which contains inferences over very fine-grained grammatical details (e.g., anaphor agreement, island phenomena), our non-ACLM models do relatively poorly compared to LTG-BERT and BabyLlama baselines. As the main difference is batch size, it is hard to speculate on any deeper reason for the lower performance. This is essentially a candidate for an "unprincipled" hyperparameter search, as it is hard to imagine what batch size specifically has to do with grammatical phenomena. Our ACLM models outperform our non-ACLM models slightly, but which ACLM models do best is not consistent over the supplement or the filtered portion of BLiMP. However, the overall consistency of outperformance of ACLM on the filtered BLiMP suggests that ACLM is having an effect.

On EWOK (Ivanova et al., 2024), which is a dataset of inferences over world knowledge, we have a completely different story. Our small batch-size non-ACLM ELC-BERT does far better than either BabyLlama or LTG-BERT. Our ACLM runs do even better than our non-ACLM ELC-BERT runs. There is no strong difference between any degree of dimensionality reduction for the surprisal space.

On GLUE (Wang et al., 2018), our non-ACLM ELC-BERT models are in the same range as the LTG-BERT and BabyLlama baselines. However, our ACLM runs are all superior to any model but the original ELC-BERT. We do not see any potential for speculation on the performance differences for the surprisal space dimension.

5 Conclusions and future work

For grammatically fined-grained inference tasks, our BLiMP results show that we underperform all models including the baseline, even without ACLM, which we would expect to be similar to the baselines or the original ELC-BERT. We can straightforwardly suggest that our ELC-BERT attempts were limited by the fact that we trained with a much smaller batch size, although the actual effect of batch size probably needs significantly more exploration, especially why BLiMP specifically is affected by the batch size issue.

The batch size difference seemed to have a major effect on EWOK and no effect on GLUE for non-ACLM models. The simplest explanation is that ELC-BERT is simply very sensitive to hyperparameters. To investigate this further, we plan to conduct a hyperparameter study, in particular considering that some of the differences between models are rather small. For example, tuning the learning rate to batch size could be an avenue for optimization, though this has yet to be explored. However, we can contextualize the batch size effect in terms of the performance of our ACLM training regimen.

Our ACLM models were trained under conditions similar to our ELC-BERT runs. Consequently, we did not expect them to actually exceed the LTG-BERT baseline on BLiMP. We found this to be true, again possibly reflecting the batch size dependence of the task. But we saw consistent improvements on EWOK and GLUE over both our ELC-BERT-only runs and the LTG-BERT baseline. These improvement were independent of the dimensionality of the surprisal space, but, in hindsight, this is unsurprising because the input length was already uniform.

In our entry from last year, we found that reversing the surprisal criterion (effectively choosing the least surprising candidates from the pool) caused a significant delay in result convergence, suggesting that this criterion has an effect, even if we did not have the right conditions to cause it to exceed a baseline BERT model in performance. We find yet again tantalizing evidence that there are conditions under which controlling the order of learning matters. EWOK is a world-knowledge-oriented dataset. We speculate that our learner-directed process may approximate an order that reflects cognitive dependencies among the human tasks—that is, the learner "fine-tunes" successively on increasingly "complex" tasks. Exploring this requires direct inspection of what learning order is actually chosen by ACLM and empirical investigation in to whether these orders might reflect developmental needs.

A similar explanation may apply to the consistently higher performance of our ACLM runs

[2]One reviewer suggests that this may partly be the result of a switch in averaging procedure in the evaluation pipeline provided by the task organizers.

Model	BLiMP suppl.	BLiMP filtered	EWOK	GLUE
ELC-BERT (original)	67.9	80.5	-	75.3
BabyLlama	59.5	69.8	50.7	63.3
LTG-BERT	60.8	60.6	48.9	60.3
ELC-BERT B32	50.1	47.9	65.2	63.4
ELC-BERT B512	47.8	49.1	64.9	61.0
ELC-BERT ACLM-D7	47.8	51.3	70.0	64.8
ELC-BERT ACLM-D32	51.1	50.7	69.8	65.7
ELC-BERT ACLM-D64	51.1	51.1	71.0	64.8
ELC-BERT ACLM-D128	50.0	51.8	72.1	63.5

Table 1: Average accuracy scores across the BabyLM evaluation task set for the official baselines, our "plain" ELC-BERT runs, and our ACLM runs over ELC-BERT. We also include the original Charpentier and Samuel (2023) result.

on GLUE. GLUE contains common-sense reasoning entailments, and this may reflect an implicitly preferable learning order that our surprisal criterion is finding.

We emerge from this task optimistic about ACLM as a way of exploring learner-directed strategies for simulating language acquisition through training large language models. There is still a huge methodological space to explore as well as many potentially relevant hyperparameters. For efficiency and comparability reasons, we adopted ELC-BERT's sentence-independent uniform input length, which likely nullified the effect of varying surprisal space dimensions. However, we believe that sentence length ought to have an effect on the learner's choices in what to focus on next. In the case of varying sentence length, the method of reduction to a uniform space would likely therefore matter and be an appropriate target of future work.

We have also focused on surprisal as the measure that steers the interactive learner, but we find it unlikely that a single measure would represent the totality of optimal behaviours. Therefore, another direction for future work would be testing other measures or combinations thereof.

Limitations

Our work is limited to tasks based in English. We do not have a full analysis of the statistical significances of the differences in the scores. There are significant areas of the model design and hyperparameter space that we did not explore. As we replaced RoBERTa with ELC-BERT for this year's BabyLM task, we lose full comparability with last year's results.

Acknowledgements

This research was funded in part by the Swedish Research Council (VR) grant (2014-39) for the Centre for Linguistic Theory and Studies in Probability (CLASP). Xudong Hong was funded by the Konrad Zuse School of Excellence in Learning and Intelligent Systems (ELIZA). We thank our student assistant Mattes Alexander Warning for searching hyperparameters for our models. We also thank the anonymous reviewers for their insightful comments.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candance Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus.

Judit Fazekas, Andrew Jessop, Julian Pine, and Caroline Rowland. 2020. Do children learn from their prediction mistakes? a registered report evaluating error-based theories of language acquisition. *Royal Society Open Science*, 7(11):180877.

Xudong Hong, Sharid Loáiciga, and Asad Sayeed. 2023. A surprisal oracle for active curriculum language modeling. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 259–268, Singapore. Association for Computational Linguistics.

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*.

Borna Jafarpour, Dawn Sepehr, and Nick Pogrebnyakov. 2021. Active curriculum learning. In *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*, pages 40–45, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Lillian R. Masek, Brianna T.M. McMillan, Sarah J. Paterson, Catherine S. Tamis-LeMonda, Roberta Michnick Golinkoff, and Kathy Hirsh-Pasek. 2021. Where language meets attention: How contingent interactions promote learning. *Developmental Review*, 60:100961.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. ArXiv preprint 1804.07461.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Gotlieb Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Adina Williams, Bhargavi Paranjabe, Tal Linzen, and Ryan Cotterell. 2023. Findings of the 2023 BabyLM Challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the 2023 BabyLM Challenge*. Association for Computational Linguistics (ACL).

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.

Zhisong Zhang, Emma Strubell, and Eduard Hovy. 2022. A survey of active learning for natural language processing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6166–6190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Pre-training details

Hyperparameter	Small (Submitted Model)
Number of parameters	24M
Number of layers	12
Hidden size is	384
FF intermediate size	1 024
Vocabulary size	6 144
Attention heads	6
Hidden dropout	0.1
Attention dropout	0.1
Training steps	31 250
Batch size	512
Initial Sequence length	128
Warmup ratio	1.6%
Initial learning rate	0.005
Final learning rate	0.005
Learning rate scheduler	cosine
Weight decay	0.4
Layer norm ϵ	1e-7
Optimizer	LAMB
LAMB ϵ	1e-6
LAMB β_1	0.9
LAMB β_2	0.98
Gradient clipping	2.0
Gradient accumulation	4

Table 2: Pre-training hyperparameters for ACLM models trained on the STRICT-SMALL track. Note that they are almost identical to the SMALL ELC-BERT model (Charpentier and Samuel, 2023), with the exception of the batch size and the gradient accumulation.

Dreaming Out Loud: A Self-Synthesis Approach For Training Vision-Language Models With Developmentally Plausible Data

Badr AlKhamissi[*] **Yingtian Tang**[*] **Abdülkadir Gökce**[*]
Johannes Mehrer[†] **Martin Schrimpf**[†]
EPFL

Abstract

While today's large language models exhibit impressive abilities in generating human-like text, they require massive amounts of data during training. We here take inspiration from human cognitive development to train models in limited data conditions. Specifically we present a self-synthesis approach that iterates through four phases: Phase 1 sets up fundamental language abilities, training the model from scratch on a small corpus. Language is then associated with the visual environment in phase 2, integrating the model with a vision encoder to generate descriptive captions from labeled images. In the "self-synthesis" phase 3, the model generates captions for unlabeled images, that it then uses to further train its language component with a mix of synthetic, and previous real-world text. This phase is meant to expand the model's linguistic repertoire, similar to humans self-annotating new experiences. Finally, phase 4 develops advanced cognitive skills, by training the model on specific tasks such as visual question answering and reasoning. Our approach offers a proof of concept for training a multimodal model using a developmentally plausible amount of data.

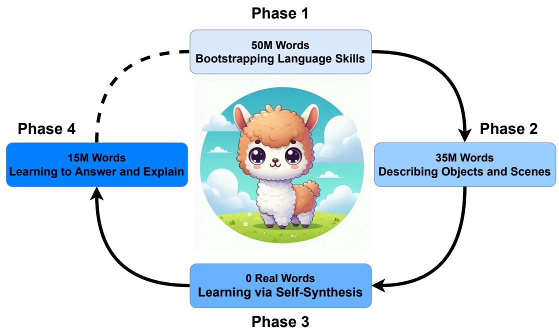


Figure 1: **Self-Synthesis Training Framework**. Our model `BabyLLaMA` is trained in four phases that connect fundamental language abilities to vision by learning to describe unlabeled visual experiences. We divided our approach in 4 phases, each feeding its best snapshot in terms of validation loss to the next phase. Phase 1 is concerned with fundamental language skill acquisition using 50M words. Phase 2 combines visual and text data (35 M words) to learn to describe objects and scenes. In phase 3 - making our approach one revolving around self-synthesis - we generate captions from images and use this synthesized text (i.e., 0 words from real-world corpora) to further train the model's language decoder. Phase 4 closes the loop using 15M words to develop skills for advanced visuo-linguistic tasks such as question answering and reasoning about the environment.

1 Introduction

Recent advances in machine learning have produced large language models (LLMs) that, after training on massive text corpora, are capable of generating human-like text. However, when comparing LLM training to human development, the amount of data required for successful model training far exceeds the quantities that humans learn from during their development (Warstadt et al., 2023a). The human brain is thus often seen as a more sample-efficient learning machine than contemporary artificial neural network approaches (Marcus, 2020).

In this work, we take inspiration from human cognitive development to build new models under limited data conditions that more closely resemble the language experience of humans. Specifically, humans learn language in combination with other senses, and use it to reflect on their experiences. We implement this idea via a *self-synthesis* approach that combines vision and language such that the model learns on external (real-world) text as well as its own (synthetic) description of unlabeled visual experiences (Figure 1). Self-synthesis can also be seen as analogous to the process of dreaming, which neuroscience research suggests functions as providing "augmented samples of waking experiences," helping to shape neural representations

[*]Equal Contribution
[†]Equal Supervision

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 244–251
November 15-16, 2024

and prevent overfitting to those experiences (Hoel, 2021; Prince and Richards, 2021).

2 Dataset Selection

In line with the BabyLM challenge requirements (Warstadt et al., 2023b), we restrict our training data to 100 million words, which approximates the maximum number of words a 13-year-old would encounter in their lifetime (Gilkerson et al., 2017). In contrast, the latest `LLaMA-3-8B` model was trained on 15 trillion tokens (Dubey et al., 2024), which is 150,000 times larger than our training budget. We created our own dataset of 100 million words, emphasizing diversity and quality. This word budget is split evenly between a text-only corpus and a multimodal image-text corpus.

Text-Only Data Our text corpus comprises 50 million words selected from the top-scoring sentences of `FineWeb-Edu`'s October 2024 CommonCrawl snapshot (Lozhkov et al., 2024), based on their educational quality. `FineWeb-Edu` is a subset of the `FineWeb` dataset (Penedo et al., 2024), which is created using scalable, automated annotations to assess educational value. The educational scores were assigned by `LLaMA-3-70B-Instruct`, which rated 500,000 samples on a scale from 0 to 5 for their educational quality (Penedo et al., 2024). Models trained on this dataset have surpassed all other publicly available web datasets on several educational benchmarks, including MMLU (Hendrycks et al., 2021), ARC (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018).

Image-Text Data Our image-text corpus consists of two groups: (1) image-caption data used for visual experience training ("phase 3" Section 5.3); (2) multi-task image-text data used for finetuning the model towards advanced reasoning ("phase 4", Section 5.4), which include captioning, VQA, and visual reasoning. For the images with captions used for visual experience training, we select subsets from `WIT` (Srinivasan et al., 2021), `obelics` (Laurençon et al., 2024), and `LAION` (Schuhmann et al., 2021). These datasets include diverse image descriptions such as wikipedia paragraphs, news, and also simple short captions. We sampled 27 million, 5 million, and 3 million words respectively from the 3 datasets. For the multi-task image-text data, we used `M3IT` (Li et al., 2023), a dataset curated for multi-lingual instruction tuning and sampled 15 million words from it. The goal is to enhance the model's ability to follow instructions as well as gain more advanced skills such as visual-reasoning, such that it can utilize its acquired knowledge more effectively. Taken together, the two groups of image-text data make up a total of 50 million words. The selection of these datasets was not arbitrary; it resulted from multiple iterations aimed at ensuring both diversity and quality.

3 Benchmarks

We evaluate our model across six benchmarks: three focused on language-only tasks and three on vision-language tasks. Except for GLUE, where we fine-tune the model on each subtask using LoRA (Hu et al., 2022), all benchmarks are evaluated in a zero-shot setting.

3.1 Language-Only Benchmarks

BLiMP BLiMP is a benchmark that evaluates key grammatical phenomena in English. It is composed of 67 sub-datasets, each containing 1,000 minimal pairs designed to highlight specific contrasts in syntax, morphology, or semantics. The data is automatically generated based on grammars developed by experts (Warstadt et al., 2019).

Elements of World Knowledge (EWoK) EWoK is a benchmark that evaluates the world modeling abilities of language models. It covers 11 key domains of world knowledge essential for human-like world modeling. These domains range from reasoning about spatial relations to understanding social interactions (Ivanova et al., 2024).

GLUE The General Language Understanding Evaluation (GLUE) benchmark is a comprehensive suite of resources designed to train, evaluate, and analyze natural language understanding models. It includes nine diverse tasks focused on sentence or sentence-pair understanding, drawn from well-established datasets. These tasks vary in dataset size, text genre, and complexity, providing a broad assessment of language understanding capabilities (Wang et al., 2018). In our experiments, we utilize LoRA (Hu et al., 2022), a parameter efficient fine-tuning method, in order to tune our model to the GLUE tasks.

3.2 Vision-Language Benchmarks

VQA We use the second version of the Visual Question Answering (VQA) benchmark that builds upon the original VQA (Zhang et al., 2015) by incorporating complementary images. In this dataset,

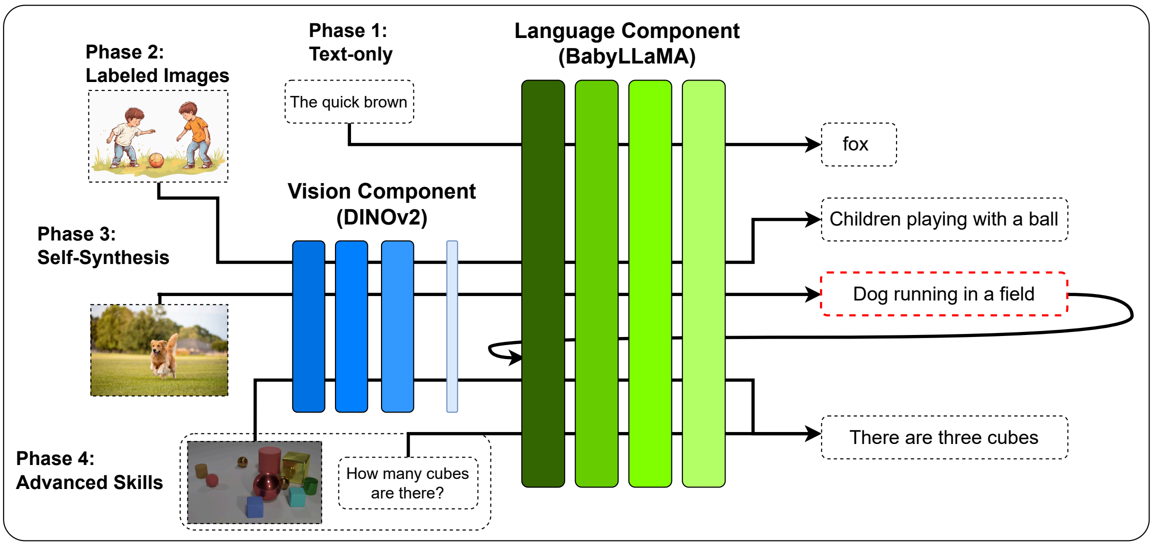


Figure 2: Overview diagram illustrating the four phases of training. Starting from training on text only (phase 1), language capabilities are connected to images (phase 2). The model then self-synthesizes text (red border) on unseen images, and uses this text to continue training the language component (phase 3), which is further refined for e.g. question answering (phase 4). Sizes of model components do not reflect number of parameters.

each question is linked to a pair of similar images, each yielding a distinct answer, thus increasing the challenge. For the model to answer these questions, it requires a grasp of vision, language, and commonsense knowledge (Goyal et al., 2016).

Winoground Winoground is a challenging task and dataset designed to assess the visio-linguistic compositional reasoning abilities of vision-language models. The objective is to correctly match two images with two captions, where both captions use the exact same words or morphemes but arranged in different orders. Expert annotators carefully curated the dataset, providing fine-grained tags to facilitate a detailed analysis of model performance (Thrush et al., 2022).

DevBench This benchmark contains 7 tasks across lexical, syntactic, and semantic domains, each accompanied by human response data at the item level, allowing for detailed comparisons between model scores and human response distributions. The lexical tasks evaluate vocabulary knowledge by assessing the model's ability to correctly identify the visual referent of a given noun. Syntactic tasks test grammatical understanding, requiring the model to choose the correct scene that aligns with a provided sentence. Semantic tasks measure the model's ability to represent conceptual similarity, either visually or linguistically, by comparing representational similarity scores (Tan et al., 2024).

4 Model Details

We use the same model architecture provided by the BabyLM Challenge organizers, called `BabyLLaMA`, which consists of a reduced `LLaMA` architecture, and we equip it with the `DINOv2`$_{\text{Large}}$ vision encoder to be able to handle visual inputs.

Tokenizer We train a BPE tokenizer with a vocabulary size of 16,000 on the text data from the curated dataset described in Section 2.

Language Model The language models employs the same architectural components as the `LLaMA` model (Dubey et al., 2024), but with only 16 Transformer layers and a reduced hidden dimension size of 512. The intermediate size in the MLP is 1,024, resulting in a total model size of 58 million parameters. The attention mechanism uses 8 attention heads in each layer.

Vision Encoder To equip our language model with visual capabilities, we incorporate the pretrained `DINOv2`$_{\text{Large}}$ vision encoder (Oquab et al., 2023). `DINOv2` was trained on a large-scale, unlabeled image dataset. It is built on the ViT architecture (Dosovitskiy et al., 2020) and generates 256 vision tokens per image. The vision encoder remains frozen during all experiments to preserve its pretrained features.

Projection Module The projection module serves as the bridge between the vision encoder

Image	Synthetic Description
	Hot off the field at the Ravensboro Golf Club in Ravensboro, IL. I am looking forward to the win of the season. I love the game and hopefully the games are really going to be a big thing...
	The first section of the East End of London's West End was dedicated to the Holy Spirit. The West End of London's West End was the last part of the East End of London...
	The airport is in the midst of a multi-year, $10 billion contract with the U.S. Navy, which is expected to be operational over the next few years. The agreement is expected to be signed by the United States, Canada, and the United States...

Table 1: Synthetic descriptions generated by the model for the images shown. This table illustrates the model's ability to associate visual cues with corresponding textual representations.

and the language model. It comprises a two-layer MLP with a GeLU activation function in between. This module projects the concatenated image tokens to match the dimensionality of the language model and is learnable throughout the training process.

5 Self-Synthesis Training Phases

Our framework trains the model in four phases. In each phase, we record the model checkpoint with the lowest validation loss and use it as a starting point for the following phase. For all phases, we use the AdamW optimizer combined with a cosine learning rate scheduler and a batch-size of 256. The learning rate begins with a linear warm-up phase and then gradually decreases to zero over the course of the training.

5.1 Phase 1: Bootstrapping Language Skills

Similar to how children learn a fundamental linguistic repertoire with supervision from their environment, the language component of our model is first trained from scratch on a text-only corpus. Specifically, we train `BabyLLaMA` for 15 epochs on fewer than 50 million words, using the top-scoring sentences from `FineWeb-Edu` based on their educational quality. Rather than concatenating and chunking the entire corpus into the maximum sequence length, as is common in language model pretraining, we divided each document from the `FineWeb-Edu` snapshot into individual sentences. Each sentence was truncated to have a maximum of 256 tokens and a minimum of 10 tokens. We found that training on shorter sequences by segmenting documents in this way resulted in better performance on the `BLiMP` benchmark (Warstadt et al., 2019) compared to training with fixed long sequences. The model was trained with a peak learning rate of $1e-4$ and a linear warm-up for the first $5,000$ optimization steps. (Learning rates $1e-4, 5e-5, 1e-5$ were tried and the one with the lowest validation error was chosen. We did not conduct other hyperparameter selections due to the limited resources. This also applies to other phases.)

5.2 Phase 2: Learning to Associate Language and Vision

Inspired by children learning to associate words with the objects they encounter daily, this training phase integrates a $\texttt{DINOv2}_{\text{Large}}$ vision encoder into the model to link visual inputs with language. The model is trained on image-text pairs, keeping the weights of the vision encoder frozen. We first divide each image into 16x16 patches. These 256 tokens are then transformed into feature embeddings by the model. We concatenate every 4 consecutive tokens together to form one embedding to reduce the number of tokens from 256 to 64 before passing them to the projection module. Training involves an autoregressive loss applied exclusively to the text tokens, conditioned on the corresponding image embeddings. In this setup, the projected image embeddings are concatenated with the text embeddings $\mathbf{t}_{1:s}$ before being passed through the language model. This allows the model

Phase	*Language-Only Benchmarks* BLiMP	BLiMP Supp.	EWoK	GLUE	*Vision-Language Benchmarks* VQA	Winoground	DevBench
Phase 1	0.723	0.533	0.500	0.651	-	-	-
Phase 2	0.728	**0.561**	0.504	0.650	0.395	0.507	0.242
Phase 3	**0.736**	0.556	**0.514**	0.647	0.380	0.507	**0.350**
Phase 4	0.729	0.542	0.502	**0.659**	**0.420**	**0.509**	0.228

Table 2: Performance comparison of the model across different phases of training on various benchmarks. The results show accuracy scores on language-only benchmarks (BLiMP, BLiMP Supp., EWoK, GLUE) and multimodal tasks (VQA, Winoground, DevBench). All benchmarks are evaluated in a zeroshot manner, except for GLUE, which is first finetuned using LoRA for each of its tasks separately. The best result across phases is highlighted in **bold**.

to learn a joint representation that conditions the text generation on the visual context provided by the image.

Formally, let $\mathbf{i} = \{i_1, i_2, \ldots, i_{64}\}$ be the set of image embeddings produced by the vision encoder for a given image, and $\mathbf{t} = \{t_1, t_2, \ldots, t_s\}$ be the sequence of text tokens associated with that image, where $s \leq 512$. The training objective is to maximize the conditional likelihood of the next text token t_{s+1} given the projected image embeddings and the preceding text tokens, where f is the projection module. This can be formulated as:

$$\max_{\theta,\phi} \sum_{n=1}^{N} \sum_{s=1}^{|\mathbf{t}_n|} \log p_{\theta,\phi}\left(t_{n,s+1} \mid [\, f(\mathbf{i}_n); \mathbf{t}_{n,1:s} \,]\right)$$

where: $p_{\theta,\phi}(\cdot)$ is the probability distribution generated by the combined model, $f(\mathbf{i}_n)$ represents the image embeddings processed through the projection module, $\mathbf{t}_n = \{t_1, t_2, \ldots, t_s\}$ are the text tokens for the n-th image-text pair, N is the total number of training examples, and $|\mathbf{t}_n|$ is the length of the n-th text sequence.

Therefore, just as children learn to describe their visual environment based on supervisory signals (e.g. parents describing the surroundings), the model learns to generate captions for images, articulating what it “sees.” To achieve this, we train the model to produce detailed descriptions across a diverse range of images. Consequently, we balanced the datasets to include samples with detailed descriptions (from WIT and obelics; 35842 samples / 6M words, 135393 samples / 21M words) alongside those with concise captions (from LAION; 323929 samples / 3M words). It is worth noting that although LAION contains only 3 million words, it accounts for more than half of the images due to its short captions. In this phase, we train the model for 5 epochs, with a learning rate that linearly warms-up to 10^{-5} for 250 steps, then decreases to zero throughout training.

5.3 Phase 3: Learning via Self-Synthesis

Self-Synthesis Using Images in the Wild. Beyond supervised learning on images, children also imagine and narrate stories about what they have seen. We implement this idea by having the model generate text from a set of unlabeled images and synthesizing captions that are then used to further train the language component with more diverse text. Concretely, we collected 1.1 million images from obelics that were not used during training. Using nucleus sampling (p=0.95) and top-k sampling (k=50) with a temperature of 0.7, we generated a total of 42 million words. For each image, a maximum token length between 32 and 64 was uniformly sampled. Table 1 shows a few examples of images and their corresponding text generated by our model. To avoid repetition in the generated text, we limit the maximal number of generated tokens to be 256. Note that some descriptions do not perfectly match the content of the images. This is insofar not an issue, as grammatically and vocabulary-rich text suffices for our purpose.

Continuing Pretraining Inspired by humans mixing real and imagined experiences to enhance their understanding, we train BabyLLaMA on a mixture of self-synthesized text and previously seen "real-world" data to deepen its language abilities. Specifically, we transition back from image-text training to text-only training, combining all the text data we have gathered thus far. This results in a total of 85 million real words and 42 million synthetic words. Our model is trained for just 2 epochs, with a learning rate that linearly warms up to 1e-5 over 500 optimization steps then decreases towards zero.

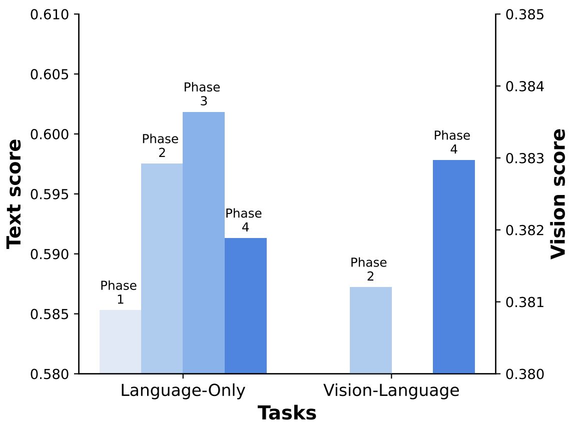


Figure 3: Average performance on all language-only (left) and vision-language-benchmarks (right) across training phases. Each phase yields a small boost for its respective training objective.

To assess the contribution of the self-synthesized text, we train another model version using only the 85 million real words and report the results on the text benchmarks in Section 6.1.

5.4 Phase 4: Learning to Answer and Explain

Equipped with fundamental language skills and the ability to describe their surroundings, human cognitive development includes answering questions and reasoning about their environment. Similarly, we train `BabyLLaMA` to handle complex visual-linguistic tasks: We finetune the language model along with the projection layer on `M3IT`. We set the learning rate to 10^{-5} with 250 warm-up updates. The model is trained for 2 epochs.

The division in 4 training phases is inspired by language acquisition in human infants. However, we do not suggest that the exact same phases accurately describe human linguistic development. For example, humans are unlikely to establish fundamental language skills (phase 1) without concurrent visual input that our model only encounters in phase 2.

6 Results

Table 2 presents the performance across various benchmarks, including both language-only and vision-language datasets. For language-only benchmarks, the phase 3 model significantly outperforms earlier models on BLiMP and EWoK, while the phase 4 model achieves the best results on GLUE. Notably, the phase 2 model delivers the highest performance on BLiMP Supplement, which is a smaller dataset compared to BLiMP. In vision-language benchmarks, the phase 4 model surpasses the phase 3 model on VQA and Winoground but underperforms on DevBench. Overall, models after phase 3 achieve the highest scores across most benchmarks. To emphasize performance differences across training phases, Figure 3 illustrates the average scores on various benchmarks. For language-only tasks, the phase 3 model shows a substantial improvement over models from phases 1 and 2. However, the phase 4 model lags slightly, likely due to fine-tuning on question-answer datasets, which shifts its focus away from general text modeling. Table 1 provides examples of synthetic descriptions generated by the phase 2 model conditioned on different images. The model accurately captures key elements in the images and produces varied syntactic and content-rich descriptions. However, there are occasional issues with logical consistency, such as the repetition of "United States" in the third example.

Benchmark	+ Synth	- Synth
BLiMP	**0.736**	**0.736**
BLiMP Supp.	**0.556**	0.550
EWoK	**0.514**	0.510

Table 3: Results of the ablation study on language-only benchmarks, comparing the performance of the model trained solely on real-world text (-Synth) against the model trained on a combination of real and synthetic data (+Synth). All benchmarks were evaluated in a zero-shot manner, illustrating the contribution of synthetic data to overall model performance.

6.1 Ablation Study

To measure the contribution of the synthetic data, we train a separate phase 3 model using only real-world text, excluding any generated text, and compare its performance with the model trained on a mixture of both real and synthetic data. Table 3 presents the results on the language-only benchmarks, all evaluated in a zero-shot manner. The findings demonstrate that incorporating synthetic data either enhances or maintains performance across benchmarks, highlighting the potential of scaling self-synthesis with larger datasets.

7 Conclusion

This work proposes a novel self-synthesis approach to training vision-language models in a data-efficient manner inspired by human cognitive development. By structuring the learning process into four distinct phases—beginning with foundational language abilities, integrating vision and language, generating synthetic data through unlabeled image captioning, and advancing cognitive tasks—the resulting model is able to solve both vision-language and language only benchmarks using a limited amount of data in a unified manner.

While we observed improved performance from each phase of training, these improvements were comparatively small. Curriculum learning methods or architectural modifications might further improve the model's learning efficiency within the proposed framework. For instance, the phases could be ran repeatedly, such that the model iteratively trains on a mix of real-world text and continuously improving self-synthesized text. A layer-fusion approach could better utilize intermediate layer representations, which has been shown to enhance training in data-limited settings (ElNokrashy et al., 2024). These efforts could close the performance gap while maintaining the developmental plausibility of the training setup. In summary, results presented here suggest that self-synthesis can make effective use of information across modalities, and might help to train performant models with developmentally plausible data regimes.

References

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, and et al. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Muhammad ElNokrashy, Badr AlKhamissi, and Mona Diab. 2024. Depth-wise attention (DWAtt): A layer fusion method for data-efficient classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4665–4674, Torino, Italia. ELRA and ICCL.

Jill Gilkerson, Jeffrey A. Richards, Steven F. Warren, Judith K. Montgomery, Charles R. Greenwood, D. Kimbrough Oller, John H. L. Hansen, and Terrance D. Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *International Journal of Computer Vision*, 127:398 – 414.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Erik Hoel. 2021. The overfitted brain: Dreams evolved to assist generalization. *Patterns*, 2(5):100244.

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

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

Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. 2024. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36.

Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, et al. 2023. M ^{3} it: A large-scale dataset towards multi-modal multilingual instruction tuning. *arXiv preprint arXiv:2306.04387*.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu.

Gary F. Marcus. 2020. The next decade in ai: Four steps towards robust artificial intelligence. *ArXiv*, abs/2002.06177.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Maxime Oquab, Timoth'ee Darcet, Théo Moutakanni, Huy Q. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Huijiao Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2023. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Preprint*, arXiv:2406.17557.

Luke Y. Prince and Blake A. Richards. 2021. The overfitted brain hypothesis. *Patterns*, 2(5):100268.

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 2443–2449.

Alvin Wei Ming Tan, Sunny Yu, Bria Long, Wanjing Anya Ma, Tonya Murray, Rebecca D. Silverman, Jason D. Yeatman, and Michael C. Frank. 2024. Devbench: A multimodal developmental benchmark for language learning. *ArXiv*, abs/2406.10215.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. 2022. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023a. 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, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell, editors. 2023b. *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Singapore.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2019. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2015. Yin and yang: Balancing and answering binary visual questions. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5014–5022.

BabyLM Challenge: Exploring the Effect of Variation Sets on Language Model Training Efficiency

Akari Haga[1] **Akiyo Fukatsu**[2] **Miyu Oba**[1] **Arianna Bisazza**[3] **Yohei Oseki**[2]
[1]Nara Institute of Science and Technology
[2]The University of Tokyo
[3]University of Groningen
{haga.akari.ha0, oba.miyu.ol2}@is.naist.jp
{akiyofukatsu, oseki}@g.ecc.u-tokyo.ac.jp
a.bisazza@rug.nl

Abstract

While current large language models have achieved a remarkable success, their data efficiency remains a challenge to overcome. Recently it has been suggested that child-directed speech (CDS) can improve training data efficiency of modern language models based on Transformer neural networks. However, it is not yet understood which specific properties of CDS are effective for training these models. In the context of the BabyLM Challenge, we focus on Variation Sets (VSs), sets of consecutive utterances expressing a similar intent with slightly different words and structures, which are ubiquitous in CDS. To assess the impact of VSs on training data efficiency, we augment CDS data with different proportions of artificial VSs and use these datasets to train an auto-regressive model, GPT-2. We find that the best proportion of VSs depends on the evaluation benchmark: BLiMP and GLUE scores benefit from the presence of VSs, but EWOK scores do not. Additionally, the results vary depending on multiple factors such as the number of epochs and the order of utterance presentation. Taken together, these findings suggest that VSs can have a beneficial influence on language models, while leaving room for further investigation.

1 Introduction

While current language models (LMs) demonstrate outstanding performance in a range of linguistic and reasoning tasks, there is ample scope to enhance their data efficiency. A state-of-the-art LM like Chinchilla uses as much as 1.4 trillion words for pretraining, whereas humans master their native language by hearing less than 100M words by the age of 13 (Warstadt and Bowman, 2022).

Child language acquisition could provide insights into it, given that children acquire basic grammar by the age of six (Paul, 1981; Kemp et al., 2005), without as varied and abundant linguistic inputs as those given to modern LMs. Various studies argue that this highly efficient learning is aided by children's limited cognitive abilities and specific types of inputs towards children (Newport, 1990; Fernald, 1985; Jusczyk, 1997; Rowe, 2012; Kempe et al., 2024). Inspired by this, the BabyLM Challenge aims at improving data efficiency in language models as well as providing insights into child language acquisition.

It is also suggested that CDS is a preferable domain for facilitating the acquisition of linguistic knowledge compared to other domains of data. The findings of these studies include efficient pretraining without sacrificing the performance (Huebner et al., 2021), enhanced semantic extraction (You et al., 2021), and superior induction of hierarchical structure (Mueller and Linzen, 2023). While these studies suggest that CDS helps LMs learn from limited datasets, further research is needed to determine which specific properties of CDS provide an advantage to LMs.

As one of such properties, some studies highlight Variation Sets (VSs), which are sets of (mostly consecutive) utterances expressing a similar intent with slight variations in the use of words and structures (Küntay and Slobin, 1996). This specific pattern is ubiquitous in CDS, but not in other speech genres. In first and second language acquisition, several studies indicate that VSs in CDS support learning of syntactic structure (Hoff-Ginsberg, 1986; Brodsky and Waterfall, 2007; Onnis et al., 2008) by maintaining children's attention on the circumscribed topic and promoting comprehension by introducing new information (Lester et al., 2022). These findings suggest that VSs are beneficial for language learning in general and thus could enhance the learning process in LMs.

In this work, we explore this hypothesis by examining the effect of VSs on language models' data efficiency. To fully control the impact of VSs, we construct artificial VSs based on the description by Küntay and Slobin (1996), mixing it with actual

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 252–261
November 15-16, 2024

CDS at various rates (0%, 20%, 40%, 60%, 80%, 100%). Then we compare the models' accuracy on these constructed datasets and shuffled datasets on BLiMP (Warstadt et al., 2020), EWOK (Ivanova et al., 2024), and GLUE (Wang et al., 2018).

2 Related Work

2.1 Child-directed Speech

CDS is a specific speech genre that parents and other caregivers use to address children, and that differs from adult-directed speech (ADS). CDS usually has simpler sentence structures, more repetitive speech, and more limited vocabulary (Snow, 1972; Farwell, 1975; Fernald et al., 1989; Kirchhoff and Schimmel, 2005).

Studies in child language development suggest that this specific speech genre is necessary for successful language acquisition among children. For example, Fernald (1985) tests 48 four-month-old infants on operant auditory preference procedure and finds that they preferred CDS to ADS. Jusczyk (1997) reports that infants can segment speech better when they hear CDS than ADS. Rowe (2012) conducts a longitudinal study on 50 parent–child dyads, demonstrating that parents' sophisticated vocabulary and decontextualized (narrative) conversation accelerate later vocabulary development in children.[1]

Following the BabyLM setup, we do not work with speech but with textual transcriptions of CDS. While sacrificing the richness of the speech signal, this choice makes the task accessible to a wider audience of computational linguistics researchers, by reducing the data complexity of the input. Henceforth, we will use CDS to denote textual transcriptions of child-directed speech.

3 Computational Studies on CDS

Computational studies further investigate whether CDS is beneficial for acquiring grammatical knowledge in models as well as for human language acquisition. Huebner et al. (2021) demonstrate that the use of child-directed speech (CDS) enables a small-sized RoBERTa (Liu et al., 2019) model trained on 5M words to attain similar linguistic competence as a RoBERTa trained on 30B words. You et al. (2021) examine that CDS has rich semantic information for grasping causal semantics without syntactic structures, finding that CDS is effective in learning to extract semantic information. Furthermore, Mueller and Linzen (2023) argue that LMs can induce hierarchical structures better when trained on CDS than on other typical datasets like Wikipedia.

While these findings demonstrate the positive effect of CDS on language learning, we are interested in which specific properties of CDS contribute to this effect. One of the reasons why CDS can enhance LMs' acquisition of syntactic structures could be its lower lexical complexity, i.e., fewer word types (Mueller and Linzen, 2023), which stems from the high repetitiveness of items in CDS. However, this repetition occurs across multiple utterances, a characteristic unique to CDS. We hypothesize that this could be a key factor in the success of LMs' language learning.

3.1 Variation Sets

In studies of first and second language acquisition, VSs (Küntay and Slobin, 1996) have gained attention as a key factor in successful language development. Küntay and Slobin (1996) describe the characteristics of VSs as follows: in successive utterances, 1) the same content is repeated or rephrased, 2) the semantic intent remains consistent, and 3) operations such as word substitution, phrase addition or deletion, and phrase reordering occur. An example of a typical VS in English is provided by Wirén et al. (2016, p.44):

(1) You can put the animals there.
You can take the pig and the cat and put them there.
Can you put them there?
Good.
Can you put the pig there too?

Several studies suggest that VSs indeed enhance language learning. For example, Hoff-Ginsberg (1986) argue that repeating identical utterances boosts children's syntactic development, while consecutive utterances with slight variations provide clues about sentence structure, aiding syntactic development. Brodsky and Waterfall (2007) conduct a corpus-based study and demonstrate that utterances with partial repetitions, such as VSs, can be overly information-dense for learners. Onnis et al. (2008) investigate the effect of VSs in language learning by teaching adults an artificial language. Their results show that VSs help adult learners parse sentences, suggesting that comparing

[1] Note that in several cultures, CDS is infrequent. (Cristia et al., 2019; Weber et al., 2017).

consecutive sentences provides clues for learning syntactic structures. Taken together, these findings suggest that seeing contextually consistent utterances with slight differences in wording could make structural differences more salient, leading to better prediction for syntactic structure in LMs.

4 Method

Inspired by human studies of language acquisition, we want to examine whether VSs can also help a language model recognize sentence structures in the language. To our knowledge, this effect has only been explored in the pilot experiment of Katano (2024). The experiment consisted of extracting naturally occurring VSs from CDS data using multiple automatic VS detection methods. The results showed no significant effect of VSs on syntactic performance measured on BLiMP, but this could be due to the difficulty in fully controlling the number of actual VSs detected automatically (Lester et al., 2022).

To address this difficulty, we opt for the use of synthetic VSs, which allows us to fully control the proportion of VSs in our training dataset. For this purpose, we use `gpt4o-mini`[2] to generate artificial VSs and augment the training data in different proportions.[3] While the use of a large LM seems to contradict the goal of improving data efficiency, we see this as a first step to measuring the importance of VSs. In case of successful results, future work could explore less costly methods to generate VSs, such as template- or syntactic rule-based.

Humans hear sentences in a VS sequentially. However, it is not clear how to present VSs to a model in a way that is equivalent to human input, nor how to maximize the effect of VSs. Therefore, we conduct experiments using two methods. The first method, as shown in the example on the left side of Figure 1, involves concatenating the VS into a single instance and providing it to the model. In this configuration, the model is forced to sequentially process the sentences within the VSs. We named this method the "Sequential Concatenation Method". The second method, as shown in the example on the right side of Figure 1, involves placing each sentence in the VS in adjacent batches. In this configuration, the model updates its parameters after processing one sentence in the VS before moving on to the next sentence within the same set. We named this method the "Adjacent Batch Method".

[2]https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

[3]To fully control the proportion of VSs within the dataset, we shuffled all sentences except for the artificial VSs to ensure that no natural VSs are included.

Original CDS	Generated VS
What do you want?	What do you need? What do you want to have? Can you tell me what you want? What is it that you want? What do you feel like getting?
What did Laura do last night?	What did Laura do yesterday evening? What was Laura doing last night? Can you tell me what Laura did last night? What activity did Laura have last night? What was Laura up to last night?

Table 1: Examples of VS generated by `gpt4o-mini`. The left column presents the original sentence in CDS, and the right column presents artificial VSs generated by the model.

4.1 Model Architecture

It has been suggested that children use predictive sentence processing, actively integrating syntactic and semantic information to foresee the upcoming categories of words (Borovsky et al., 2012). Recent studies suggest that children's predictive behavior aids their language acquisition (Reuter et al., 2019). These findings suggest that predictive processing is a powerful tool for learning sentence structure. Thus, we use GPT-2 (Radford et al., 2018), an auto-regressive (left-to-right) language model rather than a bidirectional one like BERT (Devlin et al., 2019). Hyperparameters are shown in Appendix B.

4.2 Synthesizing Variation Sets

To construct training data, we extract 10 million words of CDS from English corpora in CHILDES (MacWhinney, 2000). We eliminate utterances consisting of less than three words. In the previous literature, VSs were extracted from CDS, although these VSs contain some intervening utterances within a set of VSs:

(2) You wanna straw?
Here's your straw.
Uh oh.
Where's the straw?

Children can efficiently ignore these intervening utterances, whereas these utterances can be noisy for language models. Given that we intend

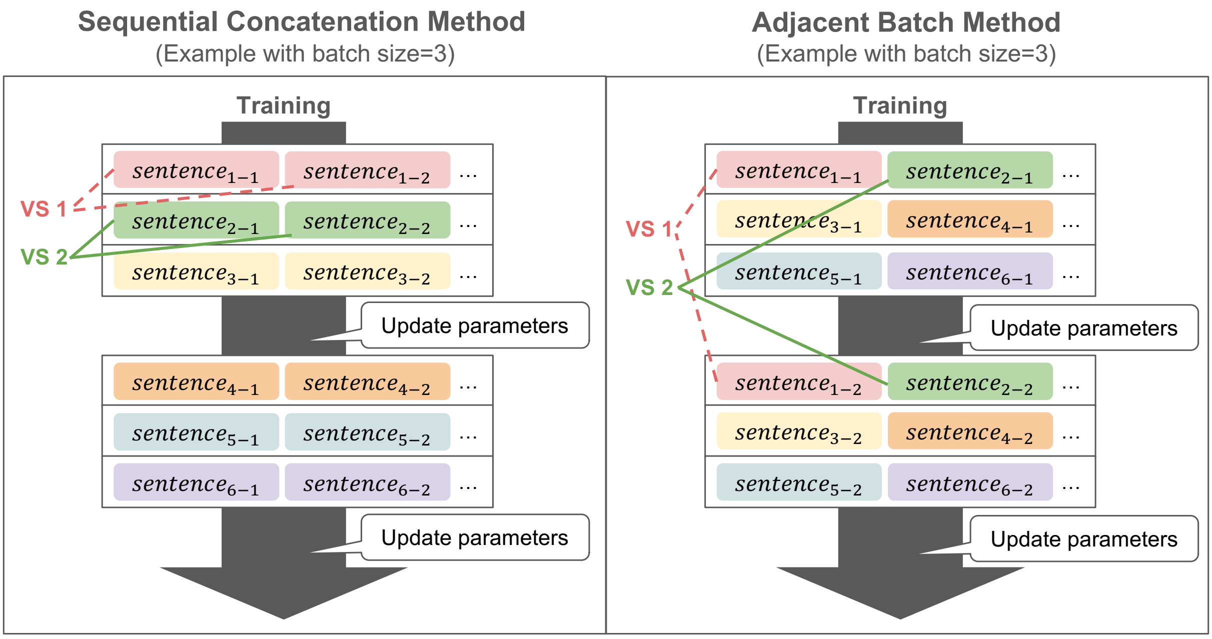


Figure 1: Two methods for inputting VSs to the model during training. Each figure illustrates an example with a batch size of 3. The figure on the left shows the method of concatenating VSs into a single sequence. In this setting, the model always processes the sentences within a VS sequentially. The figure on the right shows the method of distributing each sentence of a VS into adjacent batches. In this setting, the model updates its parameters after observing each sentence in the VS before proceeding to the next sentence in the same set. In the figures, sentence i-j indicates the j-th sentence in the i-th VS.

to explore the impact of speech patterns described by Küntay and Slobin (1996) on language models, we develop artificial data to eliminate the potential noise. For developing artificial VSs, we use `gpt4o-mini` and ask the model to generate a set of utterances that correspond to the descriptions by Küntay and Slobin (1996) and a prototypical example (see full prompt in Appendix A). Table 1 shows examples of original utterances in CDS and generated VSs based on them. Approximately 48% of the generated VSs are questions.

By using the artificial data, we can examine the upper bound of the influence of VSs on learning by language models.

4.3 Composing the Datasets

We mix artificial VSs with shuffled CDS since CDS includes a certain percentage of VS (Waterfall, 2006; Brodsky and Waterfall, 2007; Onnis et al., 2008). The percentage depends on corpora. To explore at which ratio VSs should be mixed with CDS to enhance the model's learning, we mix VSs with CDS at various ratios (0, 20, 40, 60, 80, 100). We shuffled all CDS except for the artificial VSs to ensure that no natural VSs are included. We then feed the model with these datasets using two different methods: concatenating each VS into a single sequence ("Sequential Concatenation Method") and placing each sentence in VS in adjacent batches ("Adjacent Batch Method").

4.4 Evaluation

To disentangle the effect of the *presence* of rephrasings (or variations) of the same sentence in the data from their consecutive *order* of presentation, we compare the results of each VS dataset with the shuffled version of the same dataset.

We evaluate models on BLiMP (Warstadt et al., 2020) and its supplemental tasks, EWOK (Ivanova et al., 2024), and GLUE (Wang et al., 2018) using the evaluation pipeline provided by the BabyLM organizers (Choshen et al., 2024; Gao et al., 2023). BLiMP and EWOK are used for zero-shot evaluation, whereas GLUE is used for fine-tuning evaluation. BLiMP is a binary classification task for evaluating grammatical knowledge in models and covers twelve linguistic phenomena such as agreement, binding, and island effects. EWOK aims to evaluate models' world knowledge and provides a task to match a target text with plausible or implausible contexts. GLUE provides nine different tasks, which highlight common phenomena such as the use of world knowledge, logical operation, and lexical entailment.

5 Results and Discussion

In our experiment, we train GPT-2 from scratch using training data that includes artificial VSs. We report the results after the model training has converged, specifically the results after 3 epochs.

5.1 Results Using the Sequential Concatenation Method

Table 2 shows the results for 3 epochs using the dataset containing VSs concatenated into a single line.

First, we focus on the impact of the VS ratio. In the consecutive condition, the highest macro-average score was achieved when the ratio of VSs was 0%. Although this contradicts our expectations, it may be due to reduced lexical variation in the training data caused by increased artificial VSs. Specifically, GLUE scores showed a tendency to improve as the ratio of VSs increased, whereas BLiMP scores declined with the increasing ratio of VSs. The highest BLiMP Supplement score was achieved with a 40% ratio of VSs, which aligns closely with the proportion found in real CDS, reflecting a more naturalistic distribution of CDS. The BLiMP Supplement, like BLiMP, is a binary classification task but focuses on semantic knowledge, including a question-and-answer format. Given the characteristics of this task, VSs are thought to help the model comprehend the meanings of words. VSs consist of a series of sentences with the same meaning but slightly differing in form and structure. Through these patterns, the model can recognize the meaning of each word. In contrast, the EWOK score was higher at VS ratios of 0% and 100%, which differ from the ratio found in actual CDS.

Even under the shuffled condition, scores varied with the VSs proportion. Specifically, the BLiMP score was highest at 100% VSs, while the BLiMP Supplement score peaked at 60%. The EWOK score was highest at 0%, but the difference compared to other proportions was minimal. The GLUE score peaked at 20% VSs but was nearly the same as at 0% and 100%. Overall, no consistent trend was observed in the impact of changing the VSs proportion. Next, we compare the results between the consecutive condition and the shuffle condition. In all VSs ratio settings, most scores for tasks other than GLUE were higher under the shuffled condition compared to the consecutive condition. The macro average improved by 0.89% in the shuffled condition compared to the consecutive condition. In contrast, for the GLUE scores, the consecutive condition outperformed the shuffled condition when the ratio of VSs was 60% or higher. However, a VSs proportion above 50% diverges from the actual inputs of children, as the highest proportion of VSs in CDS is approximately 50%. This discrepancy is because artificial VSs contain less noisy data compared to actual CDS. CDS contains many fragmentary utterances, as follows:

(3) To who?
You don't.
To you or to Laura?
To me.
Oh how come?

According to Cameron-Faulkner et al. (2003), fragments comprise approximately 30% of CDS. In contrast, artificial CDS contains more full sentences, as follows:

(4) It's a blanket that we all share.
We all have a blanket together.
This blanket belongs to everyone.
It's a blanket for all of us to use.
Everyone can use this blanket.

In the BLiMP Supplement, the consecutive condition outperformed the shuffled condition at a 40% ratio of VSs.

Overall, these results suggest that using training data where VSs are concatenated into a single line, VSs were effective for GLUE. While BLiMP, BLiMP Supplement, and EWOK are evaluated in a zero-shot setting, GLUE requires fine-tuning. This difference in tasks indicates that the model has not fully acquired grammatical knowledge from VSs alone. However, pre-training using VSs may enhance the efficiency of training for other tasks. However, contrary to our expectations, it is surprising that the shuffled condition, which disrupts VSs, achieved better scores.

5.2 Results Using the Adjacent Batch Method

Table 3 shows the results of 3 epochs of training using the dataset in which each sentence in the VS is placed in adjacent batches.

First, we focus on the impact of the VS ratio. Under the consecutive condition, all metrics except for

VS in Dataset	BLiMP Consec.	BLiMP Shuf.	BLiMP Suppl. Consec.	BLiMP Suppl. Shuf.	EWOK Consec.	EWOK Shuf.	GLUE Consec.	GLUE Shuf.	Macro Avr. Consec.	Macro Avr. Shuf.
0%	**60.8**	61.0	56.7	57.3	**49.9**	**50.2**	68.1	68.8	**58.9**	59.3
20%	59.0	60.6	55.8	57.5	49.1	49.5	68.7	**69.0**	58.2	59.1
40%	58.4	60.3	**58.3**	57.6	48.4	49.7	68.8	68.2	58.5	58.9
60%	57.9	60.9	55.6	**58.7**	48.7	49.6	**69.8**	68.4	58.0	**59.4**
80%	57.7	60.5	56.1	57.6	48.4	49.9	69.3	67.6	57.9	58.9
100%	57.8	**61.7**	54.8	55.4	49.3	49.6	69.6	68.8	57.9	58.9

Table 2: Averaged Scores (%) of BLiMP, EWOK, and GLUE trained on 3 epochs, where each VS is concatenated into a single sequence. Boldface denotes the highest score per benchmark in each setting. The columns Consec. and Shuf. indicate Consecutive and Shuffle, respectively.

VS in Dataset	BLiMP Consec.	BLiMP Shuf.	BLiMP Suppl. Consec.	BLiMP Suppl. Shuf.	EWOK Consec.	EWOK Shuf.	GLUE Consec.	GLUE Shuf.	Macro Avr. Consec.	Macro Avr. Shuf.
0%	60.8	61.0	56.7	57.3	**49.9**	**50.2**	68.1	**68.8**	58.9	59.3
20%	60.4	61.1	**59.7**	58.9	**49.9**	50.1	68.4	68.1	**59.6**	**59.5**
40%	60.6	60.0	58.3	**60.2**	49.3	49.5	**68.9**	67.9	59.3	59.4
60%	61.1	**61.2**	58.4	58.2	49.6	49.5	**68.9**	67.6	59.5	59.1
80%	61.5	60.8	58.8	57.8	49.6	49.5	68.6	68.0	**59.6**	59.0
100%	**61.6**	61.1	57.2	57.3	49.8	49.6	68.2	67.5	59.2	58.9

Table 3: Averaged Scores of BLiMP, EWOK, and GLUE trained on 3 epochs, where each sentence within the VS is placed in adjacent batches. Boldface denotes the highest score per benchmark in each setting. The columns Consec. and Shuf. indicate Consecutive and Shuffle, respectively.

EWOK showed better scores when the VSs were included in the training data. Specifically, the BLiMP score increased as the proportion of VSs increased. The BLiMP Supplement achieved the highest score when the proportion of VSs was 20%, which is close to the actual proportion of VSs in CDS. The GLUE score peaked when the proportion of VSs was 40% and 60%, which is slightly higher than the actual proportion in CDS. Similar to the Sequential Concatenation Method, it is likely that the increase in artificial VSs contributed to reducing noise in the training data. These results suggest that the optimal proportion of VSs varies depending on the evaluation metric. While both the BLiMP and GLUE scores benefited from the presence of VSs, the EWOK score was not affected. Under the shuffled condition, the BLiMP and BLiMP Supplement scores benefited from the presence of VSs in the training data. The highest scores for each metric were achieved when the VSs proportion was 60% or lower. The BLiMP Supplement score increased as the VSs proportion approached the human-like range of 20%–40% under the shuffle condition.

Next, we compare the results between the consecutive condition and the shuffle condition. For the scores that benefited from the presence of VSs (BLiMP, BLiMP Supplement, GLUE), the scores under the consecutive condition outperformed those under the shuffle condition when the proportion of VSs was optimal for each score.

In summary, with the Adjacent Batch Method, the consecutive condition showed higher scores for metrics other than EWOK when VSs were included in the dataset, indicating the benefit of VSs. However, the shuffled condition still outperformed the consecutive condition in some cases.

5.3 One-epoch Results

While models observe the same instances multiple times by training on multiple epochs, children only see a single instance only one time in natural speech interaction. VS has a role in increasing the salience of structural properties that are hard to recognize from a single instance, thereby exposing children to instances that have the same semantic intentions with slightly different words and structures. Therefore, there is a possibility that the effect of VSs diminishes when training over multiple epochs. We report results after training

	BLiMP		BLiMP Suppl.		EWOK		GLUE		Macro Avr.	
VS in Dataset	Consec.	Shuf.	Consec.	Shuf.	Consec.	Shuf.	Consec.	Shuf.	Consec.	Shuf.
0%	**58.3**	58.5	**54.7**	54.9	**49.4**	49.6	67.0	66.7	57.3	57.4
20%	57.5	58.8	54.6	53.8	**49.4**	49.5	68.7	68.3	57.5	57.6
40%	57.4	59.0	**54.7**	**55.2**	48.8	49.2	69.7	68.8	**57.7**	58.1
60%	57.6	**59.4**	54.1	54.9	**49.4**	49.4	69.2	**70.0**	57.6	**58.4**
80%	57.0	58.8	54.6	54.6	49.0	49.4	69.7	69.1	57.6	58.0
100%	56.6	58.7	53.3	54.9	49.0	**49.8**	**70.2**	69.2	57.3	58.1

Table 4: Averaged Scores (%) of BLiMP, EWOK, and GLUE trained on 1 epoch, where each VS is concatenated into a single sequence. Boldface denotes the highest score per benchmark in each setting. The columns Consec. and Shuf. indicate Consecutive and Shuffle, respectively.

	BLiMP		BLiMP Suppl.		EWOK		GLUE		Macro Avr.	
VS in Dataset	Consec.	Shuf.	Consec.	Shuf.	Consec.	Shuf.	Consec.	Shuf.	Consec.	Shuf.
0%	58.3	58.5	54.7	54.9	49.4	49.6	67.0	66.7	57.3	57.4
20%	58.8	59.0	55.6	55.3	**49.5**	49.6	68.3	68.7	58.1	58.1
40%	59.1	59.1	53.8	54.8	**49.5**	49.3	68.4	67.5	57.7	57.7
60%	**59.5**	**59.2**	54.1	**55.9**	49.1	49.2	68.6	**70.6**	57.8	**58.7**
80%	59.3	59.1	55.5	55.5	**49.5**	49.4	**69.2**	69.2	**58.4**	58.3
100%	58.6	58.5	**55.9**	55.4	49.2	**49.8**	68.1	68.3	57.9	58.0

Table 5: Averaged Scores of BLiMP, EWOK, and GLUE trained on 1 epoch, where each sentence within the VS is placed in adjacent batches. Boldface denotes the highest score per benchmark in each setting. The columns Consec. and Shuf. indicate Consecutive and Shuffle, respectively.

for only 1 epoch to examine this possibility.

The results for one epoch training using the Sequential Concatenation and Adjacent Batch Method are shown in Tables 4 and 5. Regarding the impact of the VSs proportion in the training data, with the Sequential Concatenation Method, the impact of VSs proportion was similar to that in the 3-epoch training: BLiMP scores decreased as the VSs proportion increased, while GLUE scores improved. The highest scores for BLiMP, BLiMP Supplement, and GLUE were observed within the 40%–60% range. With the Adjacent Batch Method, the highest scores for each metric were achieved when the VSs proportion was 60% or higher.

Regarding the differences in results between the shuffled and consecutive conditions, in the Sequential Concatenation Method results, the GLUE score was higher in the consecutive condition than the shuffled condition, except at 60% VSs. However, for most other metrics, the shuffled condition outperformed the consecutive condition. Similarly, in the Adjacent Batch Method results, none of the metrics showed a significant advantage for the consecutive condition over the shuffled condition.

Contrary to our expectations, the effects of VSs were not more pronounced in the 1-epoch training compared to the 3-epoch training.

5.4 Discussion

Taken together, our results show that the presence of CDS-inspired variations is often beneficial. However, —somewhat counterintuitively— presenting this variation in a shuffled order is often better than presenting them consecutively as in CDS. An additional finding is that the optimal amount of VSs varies among settings and evaluation benchmarks, and we could not find an overall winner. This might be due to the fact that, in our current experimental design, the amount of VSs is in direct competition with the diversity of utterances present in the dataset (i.e. potentially larger coverage of vocabulary and constructions in the datasets with less VSs). To better disentangle the effect of variations from that of corpus diversity, we are currently planning an experiment where a given amount of variations will be compared to a similar amount of identical repetitions.

6 Conclusion

We presented an initial exploration of the effect of CDS-inspired variation sets on language model training efficiency. Our results suggest that VSs can have a beneficial impact on various linguistic competences. They also reveal that this effect is entrenched with several factors like the order of utterance exposure and the number of training epochs, leaving space for more detailed investigations in the future.

7 Limitations

There are several limitations in this research. `gpt4o-mini` does not necessarily generate VSs that closely resemble natural VSs. Consequently, there is a possibility that our training data may contain unintended noise. Furthermore, we shuffled CDS to fully control the number of VSs in the training dataset. This procedure disrupted the natural VSs in CDS, possibly affecting the scores negatively. Additionally, the vocabulary size could not be strictly aligned between the "Sequential Concatenation Method" and the "Adjacent Batch Method." While the difference in vocabulary size is marginal, it may influence the scores.

8 Ethics Statement

This study was conducted in accordance with ethical guidelines and regulations. We utilized natural speech data extracted from CHILDES (MacWhinney, 2000). This is an open source corpus that archives natural speech between caregivers and their children. The data are archived without confidential information about the participants as children are usually given pseudonyms. Following the ACL Policy on Publication Ethics, we used ChatGPT to assist in refining the wording. We also partially relied on ChatGPT to generate code for prepossessing and evaluation.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 24H00087, JST PRESTO Grant Number JPMJPR21C2 and ACT-X Grant Number JPMJAX24CM. Arianna Bisazza was supported by the Dutch Research Council (NWO) within the Talent Programme (VI.Vidi.221C.009).

References

Arielle Borovsky, Jeffrey L Elman, and Anne Fernald. 2012. Knowing a lot for one's age: Vocabulary skill and not age is associated with anticipatory incremental sentence interpretation in children and adults. *J Exp Child Psychol*, 112(4):417–436.

Peter Brodsky and Heidi Waterfall. 2007. Characterizing motherese: On the computational structure of child-directed language. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Nashville, Tennessee. Cognitive Sciency Society.

Thea Cameron-Faulkner, Elena Lieven, and Michael Tomasello. 2003. A construction based analysis of child directed speech. *Cognitive Science*, 27(6):843–873.

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. *Computing Research Repository*, arXiv:2404.06214.

Alejandrina Cristia, Emmanuel Dupoux, Michael Gurven, and Jonathan Stieglitz. 2019. Child-directed speech is infrequent in a forager-farmer population: A time allocation study. *Child development*, 90(3):759–773.

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.

Carol B. Farwell. 1975. The Language Spoken to Children. *Human Development*, 18(4):288–309.

Anne Fernald. 1985. Four-month-old infants prefer to listen to motherese. *Infant Behavior and Development*, 8(2):181–195.

Anne Fernald, Traute Taeschner, Judy Dunn, Mechthild Papousek, Bénédicte de Boysson-Bardies, and Ikuko Fukui. 1989. A cross-language study of prosodic modifications in mothers' and fathers' speech to preverbal infants. *Journal of Child Language*, 16(3):477–501.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Erika Hoff-Ginsberg. 1986. Function and structure in maternal speech: Their relation to the child's development of syntax. *Developmental Pyschology*, 22(2):155–163.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

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 Akyürek, 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. *Preprint*, arXiv:2405.09605.

Peter W. Jusczyk. 1997. *The discovery of spoken language*. MIT Press.

Harufumi Katano. 2024. Syntax acquisition in language models with small datasets: Data efficiency using variation sets [in Japanese]. Bachelor's thesis.

Nenagh Kemp, Elena Lieven, and Michael Tomasello. 2005. Young children's knowledge of the "determiner" and "adjective" categories. *J Speech Lang Hear Res*, 48(3):592–609.

Vera Kempe, Mitsuhiko Ota, and Sonja Schaeffler. 2024. Does child-directed speech facilitate language development in all domains? a study space analysis of the existing evidence. *Developmental Review*, 72:101121.

Katrin Kirchhoff and Steven Schimmel. 2005. Statistical properties of infant-directed versus adult-directed speech: Insights from speech recognition. *The Journal of the Acoustical Society of America*, 117(4):2238–2246.

Aylin C. Küntay and Dan I. Slobin. 1996. *Listening to a Turkish mother: Some puzzles for acquisition*, chapter 17. Lawrence Erlbaum.

Nicholas A. Lester, Steven Moran, Aylin C. Küntay, Shanley E.M. Allen, Barbara Pfeiler, and Sabine Stoll. 2022. Detecting structured repetition in child-surrounding speech: Evidence from maximally diverse languages. *Cognition*, 221:104986.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Brian MacWhinney. 2000. *The CHILDES Project: Tools for analyzing talk. Third Edition*. Lawrence Erlbaum Associates.

Aaron Mueller and Tal Linzen. 2023. How to plant trees in language models: Data and architectural effects on the emergence of syntactic inductive biases. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11237–11252, Toronto, Canada. Association for Computational Linguistics.

Elissa L. Newport. 1990. Maturational constraints on language learning. *Cognitive Science*, 14(1):11–28.

Luca Onnis, Heidi R. Waterfall, and Shimon Edelman. 2008. Learn locally, act globally: Learning language from variation set cues. *Cognition*, 109(3):423–430.

Rhea Paul. 1981. *Analyzing Complex Sentence Development*, pages 36–71. University Park Press.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Tracy Reuter, Arielle Borovsky, and Casey Lew-Williams. 2019. Predict and redirect: Prediction errors support children's word learning. *Dev Psychol*, 55(8):1656–1665.

Meredith L. Rowe. 2012. A longitudinal investigation of the role of quantity and quality of child-directed speech in vocabulary development. *Child Development*, 83(5):1762–1774.

Catherine E. Snow. 1972. Mothers' speech to children learning language. *Child Development*, 43(2):549–565.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt and Samuel R. Bowman. 2022. What artificial neural networks can tell us about human language acquisition. *Preprint*, arXiv:2208.07998.

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.

Heidi R. Waterfall. 2006. *A little change is a good thing: Feature theory, language acquisition and variation sets*. Ph.D. thesis, University of Chicago.

Ann Weber, Anne Fernald, and Yatma Diop. 2017. When cultural norms discourage talking to babies: Effectiveness of a parenting program in rural senegal. *Child Development*, 88(5):1513–1526.

Mats Wirén, Kristina Nilsson Björkenstam, Gintarė Grigonytė, and Elisabet Eir Cortes. 2016. Longitudinal studies of variation sets in child-directed speech. In *Proceedings of the 7th Workshop on Cognitive Aspects of Computational Language Learning*, pages 44–52, Berlin. Association for Computational Linguistics.

Guanghao You, Balthasar Bickel, Moritz M. Daum, and Sabine Stoll. 2021. Child-directed speech is optimized for syntax-free semantic inference. *Scientific Reports*, 11(1):16527.

A Prompt for Generating Artificial VSs

To generate synthesis VSs, we used the following prompt:

> Rephrase a given sentence based on the characteristics of variation sets. A variation set is a set of utterances that have characteristics as follows:
> In successive utterances,
> - the same content is repeated or rephrased.
> - there is a consistent intent.
> - there are operations such as word substitution, addition/deletion of phrases, and reordering of phrases.
> Here is an example:
> You can put the animals there.
> You can take the pig and the cat and put them there.
> Can you put them there?
> Good.
> Can you put the pig there too?
> Please use only the vocabulary that 10 year-old children understand.

B Hyperparameters

Model	architecture	GPT-2
	parameters	124M
	vocab size	50,257
	hidden size	768
	heads	12
	layers	12
	dropout	0.1
	layer norm eps	1e-05
	initializer range	0.02
Optimizer	algorithm	AdamW
	learning rates	5e-05
	betas	(0.9, 0.999)
	weight decay	0.0
Scheduler	type	linear
Training	gradient accumulation	1
	epoch	3
	batch size	64
	line by line	true
	NGPU	1

Table 6: Hyperparameters of the language models.

GPT or BERT: why not both?

Lucas Georges Gabriel Charpentier[*] and **David Samuel**[*]
University of Oslo, Language Technology Group
{lgcharpe, davisamu}@ifi.uio.no

Abstract

We present a simple way to merge masked language modeling with causal language modeling. This hybrid training objective results in a model that combines the strengths of both modeling paradigms within a single transformer stack – GPT-BERT can be transparently used like any standard causal or masked language model. We test the pretraining process that enables this flexible behavior on the BabyLM Challenge 2024. The results show that the hybrid pretraining outperforms masked-only or causal-only models. We openly release the models, training corpora and code.[1]

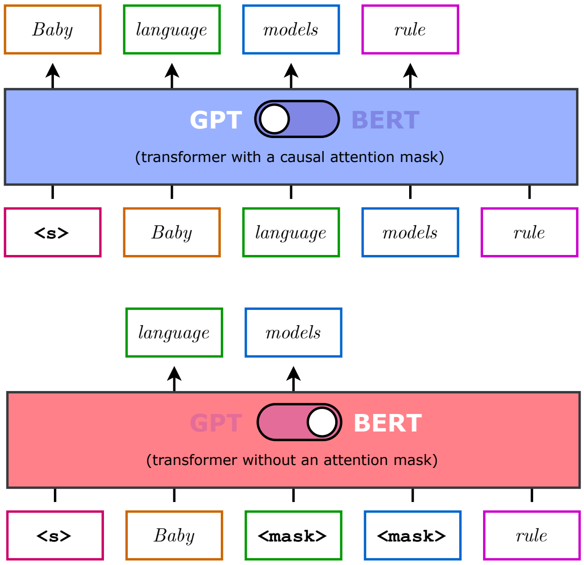


Figure 1: **Two modes of a single model** Causal and masked language modeling can be easily unified by shifting both outputs by one token to the right. Then we can train one language model on both paradigms at the same time just by modifying the input tokens, output tokens and attention masks.

1 Introduction

Language models have become fundamental tools in natural language processing, with two dominant paradigms: causal language models (CLM) and masked language models (MLM). Six years ago, GPT by Radford et al. (2018) demonstrated the generative abilities of transformer-based causal language models. Just a few months after this publication, BERT by Devlin et al. (2019) heavily outperformed the causal GPT models when finetuned on downstream NLP tasks, showcasing the major advantage of masked language modeling. These two 'historical' models define the main use-cases of the two paradigms up to this date.

The difference between these paradigms lies in how they process text. CLMs can only look at previous tokens when making predictions, mimicking the left-to-right reading process. This makes them particularly well-suited for efficient text generation. MLMs, on the other hand, can access both previous and following tokens, allowing them to build richer contextual representations. This bidirectional context has proven especially valuable for tasks requiring deep language understanding.

BERTs should not be forgotten A recent paper by Samuel (2024) revealed that BERT-like model are just as capable text generators as GPT-like models. Yet, when these two types of models are evaluated on a shared ground – generative in-context learning (Brown et al., 2020) – they still show radical differences, clearly outperforming each other in different areas. Each paradigm has its own strengths and combining them into a single hybrid might lead to a model with a more general language understanding.

GPT-BERT This motivated us to introduce GPT-BERT, a hybrid language model that combines the

[*]Both authors contributed equally to this work.

[1]The models are available on HuggingFace at `ltg/gpt--bert-babylm-base` and `ltg/gpt-bert-babylm-small`; the corpora at `ltg/babylm-2024-baby-cosmo-fine-100m` and `ltg/babylm-2024-baby-cosmo-fine-10m`. The training scripts are available on GitHub at `ltgoslo/gpt-bert`

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 262–283
November 15-16, 2024 ©2024 Association for Computational Linguistics

strengths of both CLM and MLM approaches. Our key insight is that the two objectives can be unified by reformulating how output tokens are handled in the MLM framework. Instead of predicting masked tokens at their original positions, we shift the predictions one position to the right, aligning them with the CLM's next-token prediction pattern. This simple modification allows us to train a single model that can seamlessly switch between masked and causal modes without any architectural changes or additional parameters.

This paper demonstrates the benefits of the hybrid approach across multiple benchmarks. We evaluate GPT-BERT on the *BabyLM Challenge 2024* (Choshen et al., 2024), which provides a controlled environment for comparing language models trained on limited data. Additionally, we explore the impact of varying the ratio between MLM and CLM, and we test the model's ability to perform in-context learning and text generation.

The results suggest that integrating MLM and CLM objectives during pretraining leads to more robust and capable language models, even in low-resource scenarios, without any extra training cost. Our approach opens up new possibilities for developing more efficient and versatile models for a wide range of natural language processing tasks.

2 Method

2.1 Hybrid masked-causal language modeling

In order to align both objectives we use a slightly modified version of masked language modeling called **masked next-token prediction** (MNTP; BehnamGhader et al., 2024). The only difference from traditional MLM is that when the token at position $k+1$ is masked, its prediction should be outputed at position k. In this way both MLM and CLM are unified as the output at position k always represents the token at position $k+1$. These two modes are illustrated in Figure 1.

Dataset handling To ensure that our model sees all the data for both objectives, we duplicate our dataset. One is used for the causal objective, and the other for the masked objective. We can then decide a ratio of causal-to-masked in which to divide the data seen by the model at each batch.

Loss and transformer architecture No additional changes are needed. Both training objectives minimize the cross-entropy loss, they share all learnable parameters, and use the same transformer encoder/decoder module.

2.2 Other modifications

We base the transformer architecture of our models on LTG-BERT (Samuel et al., 2023), but make some additional modifications to improve its performance. These changes are ablated in Section 4.

Attention gate Following Jumper et al. (2021), we gate the outputs of the attention operation. This is akin to the gated linear units (GLU) that have been proposed to improve the expressivity of feed-forward modules (Shazeer, 2020). This modification also simplifies the definition of the transformer architectures, now both the attention modules and the feed-forward modules can be expressed as:

```
def layer(x: tensor, layer_id: int):
    residual = x              # skip-connection
    x = layer_norm(x)         # without parameters
    g = gate(x)               # linear projection
    if layer_id % 2 == 0:     # if attention layer
        x = attention(x)      # do attention
    else:                     # else feed-forward
        x = linear(x)         # linear projection
    x = glu(x, g)             # activation (GEGLU)
    x = layer_norm(x)         # without parameters
    x = output(x)             # linear projection
    return residual + x
```

Layer weighting We further increase the expressivity of the transformer backbone by allowing each layer to select its desired combination of outputs from previous layers. This directly follows the ELC-BERT models (Georges Gabriel Charpentier and Samuel, 2023) and the later modification by Pagliardini et al. (2024) who allow any linear combination of layers instead of restricting the combination to be convex. We also make the weighting more granular by treating the attention and feed-forward modules as separate layers. With each $\alpha_{ij} \in \mathbb{R}$ being a learnable scalar, the forward pass of the resulting transformer works as follows:

```
def transformer(subword_indices: tensor):
    output₀ = embedding(subword_indices)
    for i in range(1, n_layers + 1):
        outputᵢ = ∑_{j=1}^{i} α_ij · layer(output_{j-1}, j)
    return output_{n_layers}
```

Batch-size scheduling We improve the sample-efficiency (and speed) of pretraining by linearly increasing the batch size during training (Rae et al., 2022; DeepSeek-AI, 2024). The intuition behind this method is that high-quality gradients are mainly needed at the late stages of pretraining, the initial steps can be guided by good-enough gradients from smaller batches. The maximum batch size is taken from LTG-BERT (4M tokens), but we start the training with just $1/4$ of this value, thus dividing the total number of tokens needed for training by 2.

Mask scheduling Another way to increase the sample-efficiency is to recover more unmasked tokens during training. However, Ankner et al. (2024) showed that this might be in conflict with the downstream usage of MLMs. Thus they propose to linearly decrease the masking probability throughout the training, starting with 30% and finishing with the standard 15% masking. We adopt this scheme, believing that it also reduces the impact of smaller batches at the beginning of training.

3 Pretraining and evaluation

The main purpose of this section is to evaluate if the MLM and CLM training objectives can be merged, and to evaluate the effect of this. We base the experiments on the BabyLM challenge (Choshen et al., 2024).

BabyLM challenge This shared task provides a shared ground for experiments on small-scale language modeling. Its second iteration consists of four tracks: STRICT, STRICT-SMALL, VISION and PAPER. We participate in the first two text-based tracks. There, the submissions have to be pretrained solely on a fixed number of words, 100M in the STRICT track and about 10M words in the STRICT-SMALL track. The organizers do provide a default dataset for each track, but unlike the previous edition, the participants are not limited to using it, as long as they stay under the word count limit. For the VISION track, the participants are limited to 50M words and as many images as they want. Here the goal is to create a multi-modal model. Finally, the PAPER does not require the submission of a model to the task. This track encourages contributions related to the goal of the challenge such as new cognitively-inspired metrics. As detailed in Section 3, the submissions are compared on a shared evaluation set consisting of syntactic and natural language understanding tasks.

Training corpus We pretrain both submissions on a 1 : 1 : 1 mix of the provided BabyLM corpus, on a subset of the FineWeb-Edu corpus (Lozhkov et al., 2024), and on a small subset of the Cosmopedia corpus (Ben Allal et al., 2024). The main purpose of training on this mixture is to provide the model with more factual knowledge and more diverse language.

STRICT-SMALL track (10M words)

Model	BLiMP ↑	BLiMP-S ↑	GLUE ↑	EWOK ↑
Encoder-only *(BabyLM baseline)*	60.6	60.8	60.3	48.9
Decoder-only *(BabyLM baseline)*	69.8	59.5	63.3	50.7
ELC-BERT *(2023)*	80.5	67.9	75.3	51.0
LTG-BERT *(2023)*	80.6	**69.8**	74.5	—
GPT-BERT *(ours)*	**81.2**	69.4	**76.5**	**54.6**

STRICT track (100M words)

Model	BLiMP ↑	BLiMP-S ↑	GLUE ↑	EWOK ↑
Encoder-only *(BabyLM baseline)*	69.2	66.5	68.4	51.9
Decoder-only *(BabyLM baseline)*	73.1	60.6	69.0	52.1
ELC-BERT *(2023)*	85.8	**76.8**	78.3	56.3
LTG-BERT *(2023)*	85.3	76.6	77.9	56.0
GPT-BERT *(ours)*	**86.1**	**76.8**	**81.5**	**58.4**

Table 1: **BabyLM submission scores** The final scores of our STRICT-SMALL and STRICT models submitted to the BabyLM challenge (Choshen et al., 2024). The table also includes the winner of the last year's iteration of this shared task (ELC-BERT), the baseline for our current model (LTG-BERT), as well as the baselines provided by the organizers. Results of other submission were not available as of writing this paper. Higher scores are better, the best results in each evaluation suite are boldfaced.

Pretraining process Generally speaking, we adopt the training recipe of LTG-BERT (Samuel et al., 2023), which was optimized for pretraining on another low-resource 100 million English corpus.[2] The pretraining process is the same for both tracks, except for using a smaller vocabulary and a smaller model for the STRICT-SMALL track.

As for the STRICT track, we use a BASE-sized language model with 119 million parameters. We train a case-sensitive BPE tokenizer (Gage, 1994) with a vocabulary size of $2^{14} = 16\,384$, using solely texts from the training corpus. The BASE is trained for 15 625 steps with an average batch size of 2 million tokens. The STRICT-SMALL track is tackled by a SMALL-sized language model with 30 million learnable parameters. The subword vocabulary is reduced to $2^{12} = 8\,192$ items. The training steps of the SMALL model are reduced to 7 812.

[2] https://github.com/ltgoslo/ltg-bert

The full list of hyperparameters and implementation details are provided in Appendix A.

Evaluation We utilize the language modeling benchmark suite from the BabyLM challenge (Gao et al., 2023; Choshen et al., 2024),[3] which relies on three conceptually different evaluation tasks:

1. The GLUE and SuperGLUE datasets test the ability of a pretrained model to adapt to various language understanding tasks.

2. BLiMP and BLiMP-supplement tasks test the affinity of a model towards grammatical sentences in a completely zero-shot manner.

3. EWOK is another zero-shot task. It tests the ability of a model to understand concepts such as spatial relations or physical dynamics.

We further elaborate on each of these evaluation suites in Appendix B.

4 Experiments

4.1 BabyLM submission

Table 1 shows the performance of our models against the backbone architecture of the model (LTG-BERT), as well as last year's winner on both tracks (ELC-BERT). We can see that for the STRICT-SMALL track our model outperforms last year's winner in every benchmark and is only beaten by LTG-BERT on BLiMP-Supplement by 0.4. For our submission to the STRICT track our model outperforms or matches both models (only ELC-BERT on BLiMP-Supplement matches our model). One thing to note, is that the filtration of the evaluation datasets are slightly different leading to comparisons between not exact.

For completeness, in Table 1, we also include the performance of the models provided by the BabyLM organizers (Choshen et al., 2024). The provided encoder-only models are based on LTG-BERT (Samuel et al., 2023), and the decoder-only models are based on Baby Llama (Timiryasov and Tastet, 2023). Our models clearly outperforms these baselines on all metrics, but that might be mostly attributed to their smaller pretraining budget.

[3] https://github.com/babylm/evaluation-pipeline-2024

4.2 Masked or causal?

Since our model can learn both from masked and causal examples, the question becomes, whether using a combination of both is better than using only one of the two methods during pretraining. To evaluate this, we look at the performance of models pretrained with different causal-to-masked ratios.

The main results are presented in Figure 2. We evaluate the models on four tasks that cover distinct uses: (1) BLiMP is a zero-shot linguistic-preference task that is typically better suited for masked language models (Salazar et al., 2020); (2) MNLI is a popular dataset for evaluating the finetunability of a language model, which also benefits masked language models; (3) LAMBADA, on the other hand, is a language modeling dataset mostly used to evaluate causal language models; and (4) we also directly compute the validation loss of each model. Furthermore, when applicable, each task is tested with three settings: fully-bidirectional processing (without any attention mask), unidirectional processing (with a causal mask), and partially-bidirectional processing (with a prefix mask).

The validation loss of the causal and prefix masking is calculated on the second half of the tokens of a given input sequence, where the first half of the tokens are either seen in a bidirectional fashion (prefix) or in a causal fashion (causal). For LAMDABA the entire context is seen bidirectionally for the prefix evaluation. Finally, when fine-tuning MNLI with the causal mask, we use the same tokenization as Radford et al. (2018) where a both a delimiter token is added in-between the two sentences as well as a extract token at the end of the input (two different tokens are used).

For the MNLI hyperparameters, we did a sweep on the SST-2 dataset for each model and took the best performing hyperparameters for each model and each masking (i.e. each model and masking scheme had their own hyperparameters). We sweeped over $\{1, 3, 5\}$ for number of epochs, $\{3 \cdot 10^{-5}, 5 \cdot 10^{-5}, 1 \cdot 10^{-4}\}$ for learning rates, and $\{16, 32\}$ for batch sizes.

Bidirectional results If we start by focusing on the bidirectional results, we see that the best results for all the tasks can be found for the models with a lower causal-to-masked ratio (from 1:7 to masked-only). More specifically, the 1:7 model is the best on BLiMP and LAMBADA, the best model for MNLI is 15:16, and both those mod-

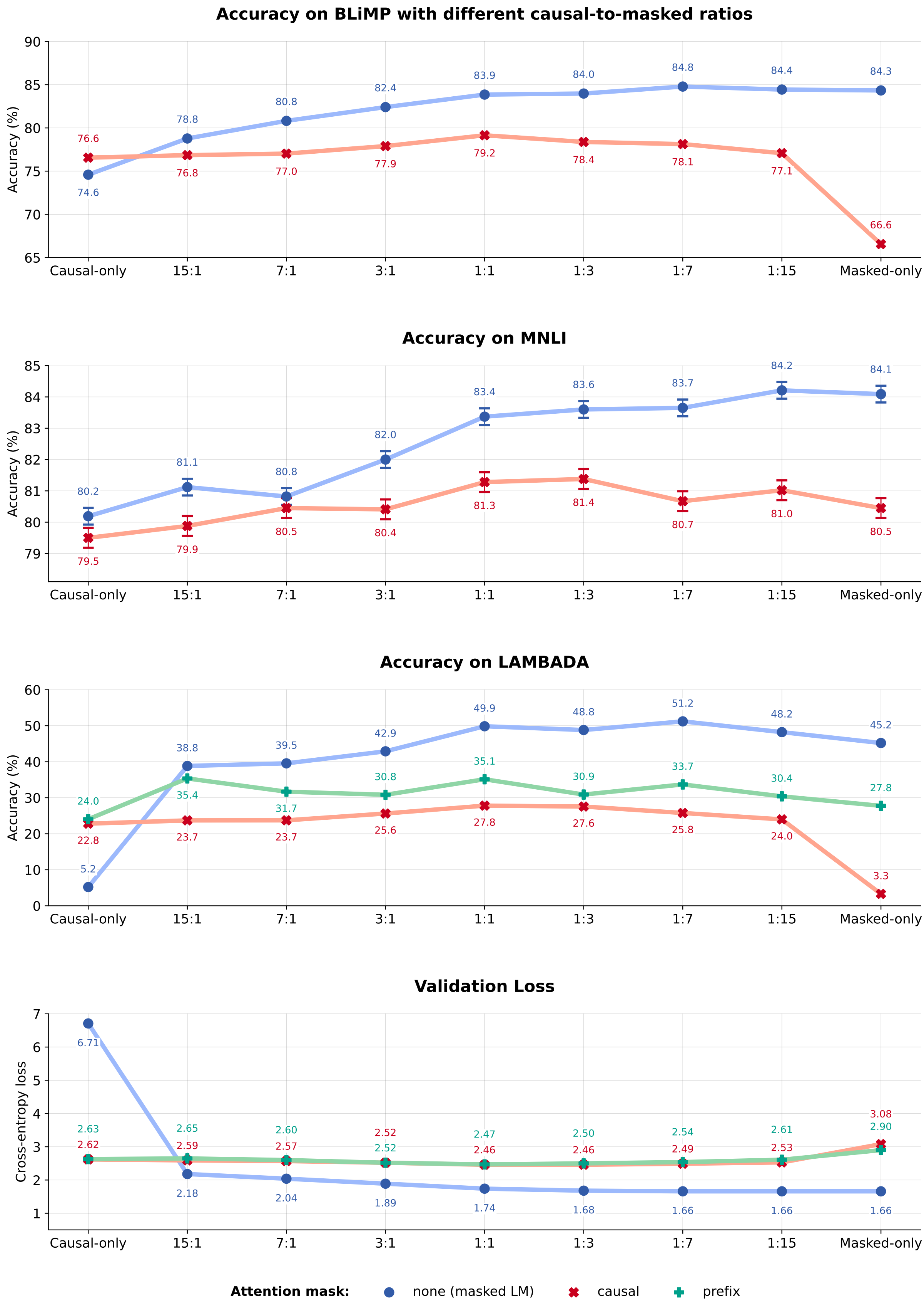


Figure 2: **The effect of the causal-to-mask ratio** Comparison of performance of different tasks when varying the ratio of MNTP used during pre-training. We also look at the performance of the model using prefix language modeling with a partially-bidirectional attention mask. MNLI scores are reported with standard deviation error bars estimated by averaging the variations across three finetuning random seeds.

els and the masked-only model achieve the best results on the validation loss. We also see that adding as few as 6.25% MNTP training can lead to significant increases in bidirectional performances (+4.2% on BLiMP, +0.9% on MNLI, +33.3% on LAMBADA and −4.53 on validation loss). In addition, using a bidirectional mask for evaluation performs the best for all models except the causal-only, however, this is unsurprising given this model is never trained to attend to every token.

Causal LM results Looking at the results when using causal masking, we see that the best models shift towards a more balanced ratio between the causal and masked training objectives. The 1:1 model and 1:3 model perform roughly the same on all tasks. As mentioned before, the results are worse than for the bidirectional evaluation; most likely because of the lower expressivity of causally-masked models (Ewer et al., 2024). Further focusing on MNLI, we see that the purely causal model does not truly benefit from being finetuned with a bidirectional mask (only +0.7% improvement, with the results being within two standard deviations of each other). Once we add some MNTP training we see a significant difference in the results between both masking strategies. With only 6.25% MNTP added, we have a 1.2% improvement when using the bidirectional mask. This trend grows to being an over 3% improvement in performance.

Prefix LM results Finally, we look at the performance for the prefix masking (partially bidirectional). We only evaluate prefix masking on LAMBADA and validation loss since it would be difficult to do this for both BLiMP and MNLI. We see that on validation loss we get similar (if not slightly worst) results as for the causal masking while the results on LAMBADA are slightly improved. In addition, the LAMBADA results do not have a clear trend outside of the hybridized models performing better than the single-objective models. This leads us to believe that our models can perform limited prefix language modeling even though they were not explicitly trained to do so.

Other benchmarks Similar trends can be seen on the other datasets in Appendix D. Based on the results on all tasks, we decided to use a 1:15 causal-to-masked ratio for our final model (to which every model is compared in subsequent sections) as well as the bidirectional evaluation scheme. In Sections 4.4 and 4.5, a model trained on this ratio is used for the in-context learning and text generation.

STRICT-SMALL track (10M words)

Model configuration	PPL ↓	BLiMP ↑	MNLI ↑	EWOK ↑
GPT-BERT	**10.8**	**81.2**	80.1	54.6
without layer weights	+0.4	−1.3	+0.2	+0.6
without attention gate	+0.3	−0.3	+0.3	−0.9
without mask scheduling	+0.1	−0.1	−0.7	−0.6
without batch scheduling	+0.7	−1.1	0.0	+**0.8**
with only BabyLM corpus	—	−0.2	−1.6	−2.0
with only FineWeb-edu	—	−0.4	+**1.1**	−0.8
with only Cosmopedia	—	−7.1	0.0	−0.6

Table 2: **Ablation study** Comparison of different model configurations proposed in Section 2.2, and corpus mixtures. The top row shows the performance of the final model (with all modifications), the middle rows show the absolute performance difference of models with one modification less, and the last group of rows shows the performance difference of GPT-BERT models trained on corpora from single sources.

4.3 Ablation study

We ablate the modeling choices from Section 2.2 as well as different choices of training data. We train the ablated models with the STRICT-SMALL setup and evaluate them on BLiMP, EWOK and MNLI (the largest GLUE dataset). The ablation results are in Table 2.

Results of the transformer ablation All our modeling decisions during development were based on the training and validation perplexities – this ablation study therefore provides an informative comparison based on a 'held-out' downstream performance. 1 In particular, the value of learnable layer weights is not clear for GPT-BERT, especially considering that they substantially slowdown the training (almost 1.5×). 2 Attention gating, on the other hand, seems to be a better substantiated improvement, which also does not add any major computational cost. 3 Mask scheduling is definitely a recommended improvement for any BERT-like models, based on all scores in this study. 4 Batch scheduling does not show on overall negative impact, which means that GPT-BERT can be trained 2.0× more efficiently using this method without a noticable degradation.

Results of the corpus ablation The ablation of the three text corpora used for training our submission shows how each of them excels in a different area – the BabyLM collection outperforms

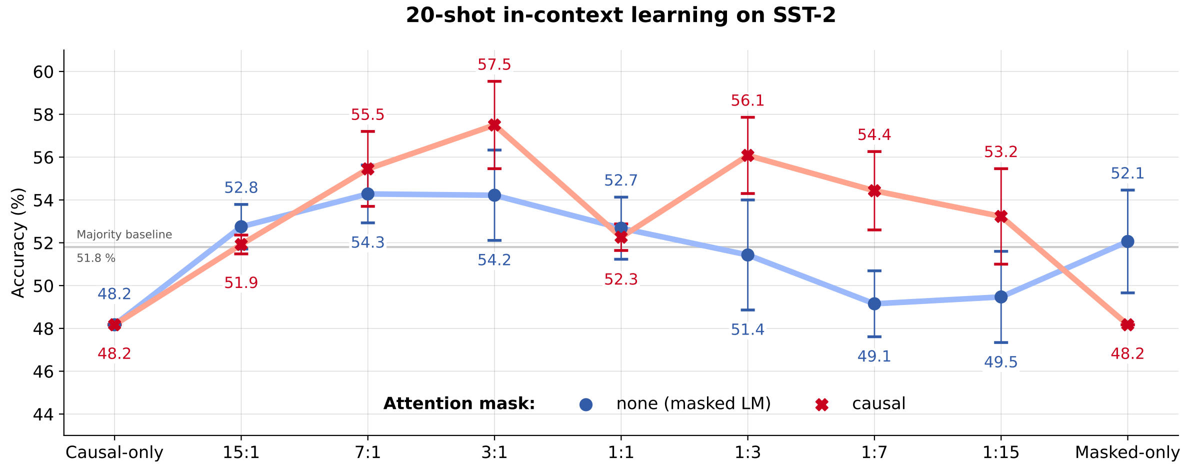


Figure 3: **SST-2 in-context learning** 20-shots ICL results on the SST-2 validation set for models trained on the 100M BabyLM datasets with varying degrees of each objective. The demonstrations (shots) were chosen at random from the training dataset. We do 20-runs and report mean as well as standard deviation. Note that the accuracy of the majority baseline on this dataset is 51.8%.

the others on BLiMP, FineWeb-Edu significantly improves the MNLI performance, and Cosmopedia is the best corpus for learning EWOK-style factual knowledge. The performance of the final GPT-BERT models shows that taking a random 1/3 of each corpus and combining them into a single collection works surprisingly well; combining the strengths of each corpus.

4.4 Text generation

None of the BabyLM benchmarks is particularly well-suited for assessing CLM performance. Thus, apart from LAMBADA, we also look at the generation capabilities of GPT-BERT. First we try standard greedy decoding with no repetition penalty (omitting new lines for readability):

> `**It was a calm night** in the small town of Harmonyville, and the residents were eager to spend the day exploring the beautiful park. Among them were two best friends, Sam and Alex, who loved to learn about nature and its wonders.`

> `**Brazil is known for** its rich history and culture. It has been home to many indigenous peoples, including the indigenous people of the Amazon rainforest. One of the most famous indigenous groups in Brazil is the Brazilian people. They are known for their strong culture, traditional beliefs, and unique traditions.`

As we can see that model is relatively repetitive, especially in the second generation, where some words are repeated in almost every sentence. If we now give the second prompt with a repetition penalty of 1.5, we get:

> `**Brazil is known for** its rich history and culture. It has been home to many indigenous peoples, including the Mayan civilization, who have built impressive cities like Tiapaca (present-day Uruguay), Cusco (now Guatemala), Chihuahua (also called Yucatán), Puebla (which was once part of Mexico), Huachimaso (then modern-day Colombia), and more. One of the most famous places in Brazil is the city of Rio de Janeiro, located on the southern tip of the Amazon basin.`

Although the model is not factually correct, it stays on topic while generating meaningful and well-formed text.

Overall, our model seems to be able to generate text, even though it has a hard time remembering exact facts and stay on topic. However, without applying repetition penalty the model struggles with repeating itself. More generations using prompts from Radford et al. (2019) can be found in Appendix C.

4.5 In-context learning

A well-known ability of larger language models is to use in-context information given in prompts to solve tasks without any finetuning – causal ones (Brown et al., 2020), as well as masked models (Samuel, 2024). However, these capabilities are often thought to appear only once a model is large enough or trained on a vast amount of data (Wei et al., 2022).

Despite the number of parameters and the size of the training corpus, our models show some signs of in-context learning, as can be seen in Figure 3.

When using the causal attention mask, we see that while the models trained with a single objective underperform the baseline, the hybrid models all perform above the majority baseline (from +0.5% to +5.7%); with the best results being achieved by the 3:1 model (with the 1:3 and 7:1 close second and third respectively). This indicates that our models are capable of doing in-context learning when trained with both objectives. When run fully bidirectionally, the trend is similar but with lower absolute performance.

5 Related work

Baby language models This paper describes a submission to the second iteration of the BabyLM challenge (Warstadt et al., 2023). Our submission is heavily inspired by the last-year's winner, ELC-BERT (Georges Gabriel Charpentier and Samuel, 2023), and by its inspiration, LTG-BERT (Samuel et al., 2023). Our modifications to these approaches are described in Section 2.1 and Section 2.2.

Hybrid masked-causal models Our work is not the first to attempt to merge bidirectional masked language modeling with generative causal modeling: T5 (Raffel et al., 2020), BART (Lewis et al., 2020) and GLM (Du et al., 2022) proposed autoregressive fill-in-the-blank training objectives, CM3 is based on a causal-mask objective (Aghajanyan et al., 2022), prefix language models use a partially-bidirectional causal modeling (Dong et al., 2019; Raffel et al., 2020), and UL2 further improves the T5 encoder-decoder with more training objectives (Tay et al., 2023). Our approach differs by its simplicity – not requiring any architectural changes nor novel training objectives – it just combines a standard causal language model with a (shifted) masked language model; the resulting hybrid can then be used as any GPT-like or BERT-like model out-of-the-box.

Masked next-token prediction To our best knowledge, this training objective was first proposed by in LLM2Vec by BehnamGhader et al. (2024), where it was used to finetune purely causal language models so that they can function as bidirectional text embedders.

6 Conclusion

We introduced GPT-BERT, a novel approach that unifies masked and causal language modeling objectives within a single transformer architecture. Through extensive experiments on the BabyLM Challenge 2024, we demonstrated that this hybrid approach offers several key advantages over single-objective models:

1. *Improved performance:* The hybrid pretraining leads to better results across multiple benchmarks, outperforming both pure MLM and pure CLM approaches.
2. *Architectural flexibility:* Without any structural modifications, our model can operate in masked, causal, or prefix modes. This flexibility enables GPT-BERT to handle a diverse range of tasks using the most appropriate inference strategy for each situation.
3. *Unexpected capabilities:* Despite being trained on limited data and having a relatively small parameter count, our models exhibit signs of in-context learning – a capability typically associated with much larger models.
4. *Training efficiency:* The hybrid approach achieves these improvements without requiring additional parameters or increased training time compared to single-objective models.

Our results suggest that the traditional dichotomy between MLM and CLM architectures may be unnecessary, and that future work might benefit from exploring more unified approaches to language model pretraining.

Limitations

While the results presented in this paper are promising and suggest improvements across many tasks when using GPT-BERT, all tested models are relatively small and trained on very small datasets. There is a possibility that these results do not scale and do not work outside of the strong BabyLM constraints.

Acknowledgments

This work is fully funded by the University of Oslo. The computations were performed on resources provided through Sigma2 – the national research infrastructure provider for high-performance computing and large-scale data storage in Norway. We acknowledge Norway and Sigma2 for awarding this project access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium through project 5000144.

References

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. 2022. CM3: A causal masked multimodal model of the internet. *Preprint*, arXiv:2201.07520.

Zachary Ankner, Naomi Saphra, Davis Blalock, Jonathan Frankle, and Matthew Leavitt. 2024. Dynamic masking rate schedules for MLM pretraining. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 477–487, St. Julian's, Malta. Association for Computational Linguistics.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second pascal recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. LLM2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

Yonatan Belinkov. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *In Proc Text Analysis Conference (TAC'09*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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. *Computing Research Repository*, arXiv:2404.06214.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

DeepSeek-AI. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *Preprint*, arXiv:2405.04434.

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.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.

Ethan Ewer, Daewon Chae, Thomas Zeng, Jinkyu Kim, and Kangwook Lee. 2024. Entp: Encoder-only next token prediction. *Preprint*, arXiv:2410.01600.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou.

2023. A framework for few-shot language model evaluation.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org.

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 Akyürek, 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. *Preprint*, arXiv:2405.09605.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, page 552–561. AAAI Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu.

B.W. Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451.

Matteo Pagliardini, Amirkeivan Mohtashami, Francois Fleuret, and Martin Jaggi. 2024. Denseformer: Enhancing information flow in transformers via depth weighted averaging. *arXiv preprint arXiv:2402.02622*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2022. Scaling language models: Methods, analysis & insights from training gopher. *Preprint*, arXiv:2112.11446.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

David Samuel. 2024. BERTs are generative in-context learners. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

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. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota. 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. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

A Pre-training details

Hyperparameter	STRICT (100M)	STRICT-SMALL (10M)
Number of parameters	119M	30M
Number of layers†	12	12
Hidden size	768	384
FF intermediate size	2 560	1 280
Vocabulary size	16 384	8 192
Attention heads	12	6
Hidden dropout	0.1	0.1
Attention dropout	0.1	0.1
Training steps	15 625	7 812
Batch size	1M $\rightarrow$ 4M (tokens)	1M $\rightarrow$ 4M (tokens)
Initial Sequence length	128	128
Final Sequence length	512	512
Warmup ratio	1.6%	1.6%
Initial learning rate	0.01	0.0141
Final learning rate	0.001	0.00141
Learning rate scheduler	cosine	cosine
Weight decay	0.1	0.1
Optimizer	LAMB	LAMB
LAMB ϵ	1e-8	1e-8
LAMB β_1	0.9	0.9
LAMB β_2	0.98	0.98
Gradient clipping	2.0	2.0

Table 3: **Pre-training hyperparameters** We train base-sized models on the STRICT corpus and small-sized models on the STRICT-SMALL corpus. † Here one 'layer' refers to one module composed of both the attention and feed-forward submodules; a more standard definition than the one used in Section 2.2.

B Evaluation details

Hyperparameters TO find the hyperparameters we do a hyperparameters search on CoLA for the task with small amounts of training data (CoLA, RTE, MRPC, MultiRC) and on SST-2 for tasks with large amounts of training data (QQP, MNLI, QNLI, BoolQ, and SST-2). We do a grid search with values:

- Number of epochs: $\{3, 5, 10\}$
- Learning rate: $\{3 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}\}$
- Batch size: $\{16, 32, 64\}$

In addition for WSC given the very low amount of both train and validation data, we expand the search to:

- Number of epochs: $\{3, 5, 10, 15, 20, 25, 30, 100\}$
- Learning rate: $\{3 \times 10^{-5}, 5 \times 10^{-5}, 7 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}\}$
- Batch size: $\{16, 32, 64\}$
- Warmup ratio: $\{0.00, 0.06, 0.15\}$

The final hyperparameters can be found in Table 4. For MultiRC, we reduce the number of epochs due to the training time.

Hyperparameter	QQP, MNLI, SST-2, BoolQ, QNLI	CoLA, RTE, MRPC	MultiRC	WSC
STRICT-SMALL				
Number of epochs	3	10	3	20
Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}	3×10^{-4}
Batch size	16	16	16	32
Warmup ratio	0.06	0.06	0.06	0.00
Weight decay	0.01	0.01	0.01	0.01
STRICT				
Number of epochs	3	10	3	20
Learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}	3×10^{-4}
Batch size	32	32	32	16
Warmup ratio	0.06	0.06	0.06	0.06
Weight decay	0.01	0.01	0.01	0.01

Table 4: **Fine-tuning hyperparameters** We use the hyperparameters above to fine-tune our models. We did a hyperparamter search on CoLA and SST-2 to obtain the hyperparameters. For MultiRC, we used less epochs due to the time required to fine-tuned.

(Super)GLUE benchmark. General Language Understanding Evaluation benchmarks (GLUE and SuperGLUE; Wang et al., 2018, 2019) are arguably the most common ways of evaluating the language-understanding and transfer-learning capabilities of language models. The BabyLM challenge uses a subset of 10 (Super)GLUE tasks, detailed in Appendix G. We employ the standard way of finetuning masked language models on these datasets, as introduced in BERT (Devlin et al., 2019).

As we use the BabyLM version of GLUE, our results cannot be directly compared with previous literature – the dataset samples are filtered to not contain out-of-vocabulary words and some of the employed metrics differ from the original recommendations (Wang et al., 2018, 2019). We opted to adhere to the BabyLM version to be compatible with other works in this challenge.[4]

BLiMP. When using any finetuning approach, it becomes unclear how to disentangle innate language understanding from knowledge learned during second-stage supervised finetuning (Belinkov, 2022). In contrast, the Benchmark of Linguistic Minimal Pairs (BLiMP; Warstadt et al., 2020) attempts to measure the linguistic knowledge of a language model in a zero-shot manner – without any additional training. Each pair of sentences in BLiMP differs minimally on the surface level, but only one of the sentences is grammatically valid. We can use the intrinsic ability of language models to assign a probability to every sentence and test how often a language model assigns a higher probability to the correct sentence (Wang and Cho, 2019; Salazar et al., 2020).

As detailed in Appendix E, the results on BLiMP greatly depend on temperature scaling (Guo et al., 2017). Thus, to fairly compare different types of language models, we employ an alternative approach to evaluating BLiMP: we report the accuracies that are achieved with the optimal temperature for every language model; the reasoning is explained in Appendix E.

The BabyLM challenge also comes with an additional 'BLiMP supplement' held-out set with five additional diagnostic tasks. To comply with the held-out spirit of these tasks, we keep the temperature values calibrated for BLiMP, even though this results in suboptimal performance (Appendix E).

EWoK. Elements of World Knowledge (EWoK; Ivanova et al., 2024) is another benchmark evaluated in a zero-shot manner. The goal of this benchmark is to evaluate how well a model understand concepts that help model the world. To do this, each example consists of two contexts, and two targets. The first context implicates the first target, and the second context implicates the second target. Both implications are based on world knowledge, such as an object both accelerating and slowing down (context) is oscillating

[4]The BabyLM pipeline unfortunately uses identical validation and test sets, which might yield overly optimistic results due to overfitting during hyperparameter optimization.

(target).

LAMBADA. LAnguage Modeling Broadened to Account for Discourse Aspect (LAMBADA; Paperno et al., 2016) is a dataset that verifies how well a language model understand context and whether it is able to recall far-away context. To do this the model must predict an answer token(s) which usually consists of a person, action, fact, or event mentioned earlier in the context.

C Text Generation with prompts from Radford et al. (2019)

Context
In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.
Completion (no repetition penalty)
The unicorns were known for their ability to communicate with each other, and they were known to have a unique way of communicating. The unicorns were known to have a unique way of communicating with each other, and they were known to have a unique way of communicating with each other. The unicorns were known to have a unique way of communicating with each other, and they were known to have a unique way of communicating with each other. The unicorns were known to have a unique way of communicating with each other, and they were known to have a unique way of communicating with each other. The unicorns were known to have a unique way of communicating with each other, and they were known to have a unique way of communicating with each other. The unicorns were known to have a unique way of communicating with each other, and they were known to have a unique way of communicating with each other. The unicorns were known to have a unique way of communicating with each other, and they were known to have a unique way of communicating with each other.
Completion (1.5 repetition penalty)
The study is published in Nature Communications. "The unicorn language has been known for centuries as an ancient communication system," said Dr. Mark Rosenberg, lead author of the paper and co-author of the new research. "It's not just a myth; it's also a real one." A group of scientists from the University of California at Berkeley (UCB) and the University of Colorado at Boulder (UCSD) have found evidence of this phenomenon in some animals. They say they've identified two species of unicorns with similar sounds, but they're different in their own way. "We are looking at how these unicorns communicate," says Dr. Rosenberg. "They use sound waves to send messages through the air, which then travel down the animal's throat, where it gets picked up by other animals, and back to the brain." Researchers believe the unicorns' ability to hear can be explained by the presence of certain chemicals called acetylcholinesterase. Acetylcholinesterase helps them detect danger, such as wind or water. It also makes them feel safe when there's a threat.

Context
Aaron loves mint chocolate cake, but he requires that it be paired with mini chocolate chips, so I threw some of those in between the layers. I also had a few Peppermint Jo Jos on hand so I crushed them up and threw some of those in along with some crushed meringue cookies because, why not? It's a total smorgasbord of minty chocolate chippy cookie crunchy goodness. I didn't measure how much of each topping I used, but after I tasted the finished product, I wish I had added more. You can add anything you want- crushed candy canes, peppermint bark, etc. And don't be afraid to use a heavy hand. Texture = good. If you don't have 7-inch cake pans, you can get 3 shorter 8-inch layers out of this
Completion (no repetition penalty)
recipe. You can also use a 1-inch layer of chocolate chips, and you can also use a 1-inch layer of chocolate chips. Now, let's talk about the texture. The texture of the cake is the amount of chocolate chips that you're using. You can use a 1-inch layer of chocolate chips, 1-inch layer of chocolate chips, and 1-inch layer of chocolate chips. You can also use a 1-inch layer of chocolate chips, 1-inch layer of chocolate chips, and 1-inch layer of chocolate chips. To make the cake, you'll need to mix 1-inch chocolate chips, 1-inch chocolate chips, 1-inch chocolate chips, and 1-inch chocolate chips. You'll also need to mix 1-inch chocolate chips, 1-inch chocolate chips, and 1-inch chocolate chips. To make the cake, you'll need to mix 1-inch chocolate chips, 1-inch chocolate chips, and 1-inch chocolate chips. You'll also need to mix 1-inch chocolate chips, 1-inch chocolate chips, and 1-inch chocolate chips.
Completion (1.5 repetition penalty)
recipe. The best part is that they are made from 120% natural ingredients. So if you need to make your own, just go ahead and buy one! Now let's talk about what makes these cakes so special. First, there are two main components: the base layer (the top) and the outer layer (the bottom). This layer consists of three layers: the base layer (which contains all the other layers), the inner layer (which includes the outer layer), and the middle layer (which has everything else). Each layer provides its unique texture and flavor profile. For example, when making a macaroon cake, we would use a thinner layer called the "middle layer" which helps hold the shape together. But if we wanted to create a thicker version, we could choose a lighter layer like a "cream cheese" or even a "dry cream cheese". Next, let's discuss the "inferior layer". This is where our final dessert comes in handy.

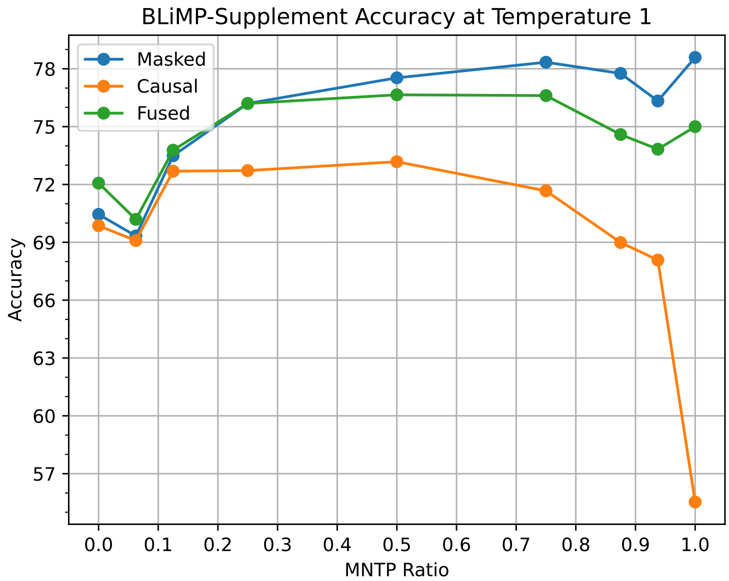


Figure 4: **BLiMP-Supplement Accuracy** Comparison of BLiMP-Supplement accuracy when varying the ratio of MNTP used during pre-training. We set the temperature to apply on the logits to 1 for fair comparison between the evaluation strategies. Fused is the sum of the logits from the causal and masked evaluation.

D Varying MNTP ratio results on other datasets

Figures 4 and 5 show the result of varying the MNTP ratio on the BLiMP-Supplement and EWoK benchmarks. We evaluate the benchmarks with the masked, causal, fused (the sum of the logits of the masked and causal scheme), and prefix (for EWoK) schemes.

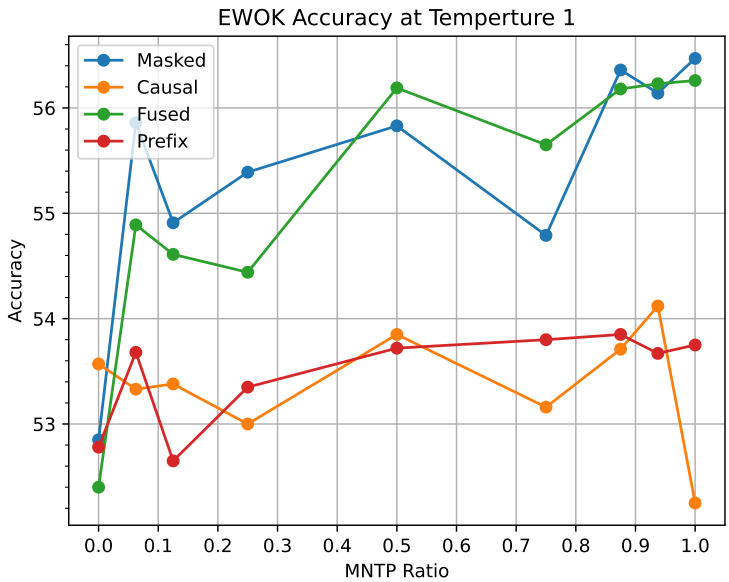


Figure 5: **EWoK Accuracy** Comparison of EWoK accuracy when varying the ratio of MNTP used during pre-training. We set the temperature to apply on the logits to 1 for fair comparison between the evaluation strategies. Fused is the sum of the logits from the causal and masked evaluation. We also look at the performance of the model using a prefix masking strategy where the whole context is visible to the model.

E BLiMP

The BabyLM challenge uses the BLiMP benchmark (Warstadt et al., 2020) to evaluate the syntactic understanding of the models. Our detailed results can be found in Table 5. The component tasks are as follows (with descriptions from Warstadt et al. (2020)):

Anaphor Agreement (AA): the requirement that reflexive pronouns like *herself* (also known as anaphora) agree with their antecedents in person, number, gender, and animacy.

Argument structure (AS): the ability of different verbs to appear with different types of arguments. For instance, different verbs can appear with a direct object, participate in the causative alternation, or take an inanimate argument.

Binding (B): the structural relationship between a pronoun and its antecedent.

Control/raising (CR): syntactic and semantic differences between various types of predicates that embed an infinitival VP. This includes control, raising, and *tough*-movement predicates.

Determiner-noun agreement (DNA): number agreement between demonstrative determiners (e.g., *this/these*) and the associated noun.

Ellipsis (E): the possibility of omitting expressions from a sentence. Because this is difficult to illustrate with sentences of equal length, our paradigms cover only special cases of noun phrase ellipsis that meet this constraint.

Filler-gap (FG): dependencies arising from phrasal movement in, for example, *wh*-questions.

Irregular forms (IF): irregular morphology on English past participles (e.g., *awoken*).

Island effects (IE): restrictions on syntactic environments where the gap in a filler-gap dependency may occur.

NPI licensing (NL): restrictions on the distribution of *negative polarity items* like *any* and *ever* limited to, for example, the scope of negation and *only*.

Quantifiers (Q): restrictions on the distribution of quantifiers. Two such restrictions are covered: superlative quantifiers (e.g., *at least*) cannot be embedded under negation, and definite quantifiers and determiners cannot be subjects in existential-*there* constructions.

Subject-verb agreement (SVA): subjects and present tense verbs must agree in number.

On temperature scaling, we observe that for the masked scheme, the increase in performance when using temperature scaling is on average of 2%. This is not the case for the causal scheme, where temperature seems to have very little effect on the performance of the model.

Model	AA	AS	B	CR	DNA	E	FG	IF	IE	NL	Q	SVA
STRICT-SMALL												
ELC-BERT*$_{(2023)}$	89.5	72.5	68.1	72.6	93.4	**87.4**	**80.6**	91.0	67.9	79.4	**75.2**	88.7
GPT-BERT	**93.6**	**78.2**	**68.8**	**77.4**	**97.3**	86.1	80.5	**91.5**	**69.8**	**84.1**	68.4	**92.2**
STRICT												
ELC-BERT*$_{(2023)}$	92.8	81.2	74.0	79.2	96.0	**91.7**	87.1	93.6	**83.9**	83.5	70.2	90.8
LTG-BERT*$_{(2023)}$	96.1	79.5	**77.1**	80.3	95.4	**91.7**	87.8	94.5	79.8	84.4	72.2	91.2
GPT-BERT	**97.7**	**84.3**	74.61	**83.7**	**98.2**	86.9	**89.3**	**96.6**	77.3	**85.2**	**76.4**	**95.1**

Table 5: Detailed BLiMP results for models trained on both tracks. The **bold** results represent the best model for the task. The metric used to measure is accuracy. The results are in percentage. *Results from (Georges Gabriel Charpentier and Samuel, 2023); they are not directly comparable due to the differences in data filtration between the models as well as the optimized BLiMP temperature being used instead of a general one.

F BLiMP Supplemental

The BLiMP Supplemental was introduced in the last version of the BabyLM Challenge (Warstadt et al., 2023). As for BLiMP it tests the syntactic understanding of models. It consists of the following 5 sub-tasks:

Hypernym Checks whether a word is a superset/subset of another word (for example a dog is a mammal so having a dog means having a mammal).

QA Congruence Easy Checks where the question type is congruent with the answer (i.e. a who question answers about a person and not a thing).

QA Congruence Tricky Same as before but with more ambiguous cases.

Subject Aux Inversion Checking whether the verb relates to the correct subject.

Turn Talking Checks whether the right personal pronoun is used in the answer to a question in a conversation.

The results can be found in Table 6.

G GLUE

The BabyLM challenge involves slightly modified GLUE and SuperGLUE benchmarks. It uses only a subset of the subtasks, the datasets are filtered so that they do not contain out-of-vocabulary words, and it sometimes uses non-standard metrics. Our detailed results can be found in Table 7. We list all subtasks and their metrics below:

Model	Hypernym	QA Cong. Easy	QA Cong. Tricky	Subject Aux Inversion	Turn Talking
STRICT-SMALL					
Encoder*(baseline)*	**54.2**	62.5	49.1	79.9	58.2
Decoder*(baseline)*	49.6	54.7	41.2	86.0	66.1
ELC-BERT**(2023)*	48.0	**73.4**	43.6	**90.0**	84.3
GPT-BERT	47.1	**73.4**	**54.5**	86.3	**85.7**
STRICT					
Encoder*(baseline)*	**55.0**	75.0	53.3	87.5	61.4
Decoder*(baseline)*	45.6	56.2	44.8	83.9	72.5
ELC-BERT**(2023)*	47.3	85.9	**63.0**	94.5	**92.1**
LTG-BERT**(2023)*	47.0	**90.6**	60.6	90.7	**92.1**
GPT-BERT	48.8	**90.6**	59.4	**96.3**	88.9

Table 6: Detailed BLiMP supplemental results for models trained on both tracks. The **bold** results represent the best model for the task. The metric used to measure performance is accuracy. The results are in percentage. *Results from (Georges Gabriel Charpentier and Samuel, 2023); they are not directly comparable due to the differences in data filtration between the models as well as the optimized BLiMP Supplemental temperature being used instead of a general one.

Boolean Questions (BoolQ; Clark et al., 2019), a yes/no Q/A dataset evaluated with accuracy.

Corpus of Linguistic Acceptability (CoLA; Warstadt et al., 2019) evaluated with the Matthews correlation coefficient (MCC; Matthews, 1975).

The Multi-Genre Natural Language Inference Corpus (MNLI; Williams et al., 2018). Its development set consists of two parts: *matched*, sampled from the same data source as the training set, and *mismatched*, which is sampled from a different domain. Both parts are evaluated with accuracy.

The Microsoft Research Paraphrase Corpus (MRPC; Dolan and Brockett, 2005), evaluated with both F_1-score (originally also evaluated with accuracy).

Multi-Sentence Reading Comprehension (MultiRC; Khashabi et al., 2018), a multiple choice question answering dataset, evaluated with accuracy (originally evaluated with the exact match accuracy (EM) and F_1-score (over all answer options)).

Question-answering Natural Language Inference (QNLI) constructed from the Stanford Question Answering Dataset (SQuAD; Rajpurkar et al., 2016), evaluated with accuracy.

The Quora Question Pairs (QQP),[5] evaluated with F_1-score (originally evaluated with accuracy).

The Stanford Sentiment Treebank (SST-2; Socher et al., 2013), evaluated with accuracy.

The Recognizing Textual Entailment datasets (RTE; Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009), evaluated with accuracy.

Winograd Schema Challenge (WSC; Levesque et al., 2012) evaluated with accuracy.

H EWoK

The BabyLM challenge uses a slightly modified EWoK benchmark (Ivanova et al., 2024). It tests all concepts but filters the dataset to include only examples where the words appear in the BabyLM dataset. Our detailed results can be found in Table 8. We list all concepts below:

Agent Properties Checks whether the model can recognize agent (conscious beings) properties (such as believe, choice, feeling, etc.)

[5] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

Model	CoLA	SST-2	MRPC	QQP	MNLI_{m}	MNLI_{mm}	QNLI	RTE	BoolQ	MultiRC	WSC
STRICT-SMALL											
Encoder*(baseline)*	0.0	85.1	82.2	34.2	68.9	68.9	76.5	58.3	68.8	58.5	61.5
Decoder*(baseline)*	2.2	86.2	82.0	83.6	72.4	74.2	82.8	49.6	65.0	60.1	38.5
ELC-BERT**(2023)*	–	$89.3^{\pm0.5}$	$85.0^{\pm1.8}$	$86.7^{\pm0.3}$	$79.2^{\pm0.3}$	$79.9^{\pm0.2}$	$85.8^{\pm0.4}$	$55.4^{\pm2.6}$	$69.3^{\pm2.0}$	$62.2^{\pm1.0}$	$59.0^{\pm5.4}$
LTG-BERT**(2023)*	–	$88.8^{\pm0.8}$	$82.3^{\pm0.4}$	$85.8^{\pm0.2}$	$78.0^{\pm0.2}$	$78.8^{\pm0.4}$	$85.0^{\pm0.2}$	$53.7^{\pm4.1}$	$64.8^{\pm2.1}$	$64.1^{\pm0.3}$	$60.5^{\pm1.0}$
GPT-BERT	**48.9**	**92.2**	**91.5**	**87.1**	**80.2**	**80.5**	**86.4**	**64.0**	**72.5**	**69.3**	**69.2**
STRICT											
Encoder*(baseline)*	34.6	91.5	83.1	86.7	77.7	78.1	78.2	46.8	61.7	52.6	61.5
Decoder*(baseline)*	37.3	88.3	86.8	84.5	75.6	76.2	83.1	60.4	66.1	62.1	38.5
ELC-BERT*(2023)	–	$91.9^{\pm1.1}$	$89.3^{\pm0.6}$	$88.0^{\pm0.1}$	$83.6^{\pm0.1}$	$83.3^{\pm0.2}$	$89.4^{\pm0.4}$	$60.0^{\pm2.8}$	$70.5^{\pm1.5}$	$66.2^{\pm2.2}$	$56.4^{\pm9.4}$
LTG-BERT*(2023)	–	$92.0^{\pm0.4}$	$87.4^{\pm0.7}$	$87.9^{\pm0.1}$	$83.0^{\pm0.4}$	$83.4^{\pm0.5}$	$89.1^{\pm0.5}$	$54.7^{\pm2.4}$	$68.4^{\pm0.5}$	$66.0^{\pm1.4}$	$61.4^{\pm0.0}$
GPT-BERT	**62.4**	**94.0**	**94.4**	**89.1**	**85.2**	**85.3**	**90.8**	**69.1**	**78.4**	**73.3**	**75.0**

Table 7: A subset of GLUE results (defined by the Baby LM challenge) for models trained on both tracks. All the results indicate the model accuracy for the task except for MRPC and QQP where the results are based on the F1-score of the positive class and CoLA which reports the MCC. The results are reported in percentage. The **bold** result indicates the best model for each dataset. *Results from (Georges Gabriel Charpentier and Samuel, 2023); they are not directly comparable due to the differences in data filtration between the models.

Material Dynamics Checks whether the model can recognize the dynamics (movement, fluidity, etc.) of a given material.

Material Properties Checks whether the model can recognize the properties (bounciness, hardness, etc.) of a given material.

Physical Dynamics Checks whether the model can recognize the physical dynamic (speed, buoyancy, etc.) of an object.

Physical Interactions Checks whether the model can recognize the physical interactions (attraction, collision, etc.) between objects.

Physical Relations Checks whether the model can recognize the physical relations (attached vs. connected, bigger vs. smaller, etc.) between objects.

Quantative Properties Checks whether the model can recognize amount (a lot vs. little of, enough vs. not enough, etc.) of an object.

Social Interactions Checks whether the model can recognize the social interactions (cooperate vs. compete, help vs. deceive, etc.) between agents.

Social Properties Checks whether the model can recognize the social property (boastful vs. humble, dominant vs. submissive, etc.) of an agent.

Social Relations Checks whether the model can recognize the social relations (boss vs. subordinate, colleague vs. boss, etc.) between agents.

Spatial Relations Checks whether the model can recognize the spatial relations (location, height, etc.) between agents, objects or a combination of them.

I LAMBADA

LAMBADA is a zero-shot language modeling task that focuses on resolving long-range dependencies in text (Paperno et al., 2016); we used its detokenized version from Radford et al. (2019). While it has been traditionally used for evaluating autoregressive language models, we adapt the task for masked language models. Note that this adaptation does not allow for a direct comparison with the autoregressive models. An illustrative sample from this dataset looks as follows:

Model	Agent	Material		Physical			Quantative	Social			Spatial
	Prop.	Dyn.	Prop.	Dyn.	Inter.	Rel.	Prop.	Inter.	Prop.	Rel.	Rel.
STRICT-SMALL											
Encoder*(baseline)*	50.2	51.0	45.3	42.5	49.1	**51.0**	48.1	51.7	53.4	50.6	45.3
Decoder*(baseline)*	50.5	51.7	**49.4**	54.2	50.4	50.6	53.5	50.7	50.3	49.8	46.7
GPT-BERT	**50.7**	**58.1**	48.8	**57.5**	**51.1**	49.9	**55.7**	**65.6**	**58.2**	**51.6**	**53.9**
STRICT											
Encoder*(baseline)*	50.1	55.8	50.6	**58.3**	48.9	**50.9**	53.8	51.4	50.8	53.8	51.4
Decoder*(baseline)*	50.1	55.5	50.0	57.5	51.4	50.5	**56.7**	52.7	49.7	50.0	49.0
GPT-BERT	**52.7**	**72.3**	**51.8**	50.8	**52.7**	48.3	52.5	**77.2**	**64.3**	**58.9**	**60.8**

Table 8: Detailed EWoK results for models trained on both tracks. The **bold** results represent the best model for the task. The metric used to measure performance is accuracy.

Prompt: *"Give me a minute to change and I'll meet you at the docks." She'd forced those words through her teeth. "No need to change. We won't be that long." Shane gripped her arm and started leading her to the dock. "I can make it there on my own,* ***{answer}****."*

Gold answer: *Shane*

We insert the whole tokenized prompt to the evaluated language model and replace the missing answer by k mask tokens, where k is the length of the tokenized gold answer. Then we evaluate the exact-match accuracy of predicting filling in the correct continuation and also the mean perplexity.

We also evaluate using the normal causal method implemented by Radford et al. (2019), as well as doing it with a prefix, where all the context tokens attend to each other.

What Should Baby Models Read? Exploring Sample-Efficient Data Composition on Model Performance

Hong Meng Yam
Stanford University
hongmeng@stanford.edu

Nathan Paek
Stanford University
nathanjp@stanford.edu

Abstract

We explore the impact of pre-training data composition on the performance of small language models in a sample-efficient setting. Using datasets limited to 10 million words, we evaluate several dataset sources—including child-directed speech (CHILDES), classic books (Gutenberg), synthetic data (TinyStories), and a mix of these (Mix)—across different model sizes ranging from 18 million to 705 million parameters. Our experiments show that smaller models (e.g., GPT2-18M and GPT2-44M) benefit from training on diverse datasets like Mix, achieving better performance on linguistic benchmarks. In contrast, larger models (e.g., GPT2-97M, GPT2-705M, and LLaMA-360M) perform better when trained on more complex and rich datasets like Gutenberg. Models trained on the CHILDES and TinyStories datasets underperformed across all model sizes. These findings suggest that the optimal dataset for sample efficient training depends on the model size, and that neither child-directed speech nor simplified stories are optimal for language models of all sizes. We highlight the importance of considering both dataset composition and model capacity for effective sample efficient language model training.

1 Introduction

In recent years, advancements in natural language processing have been largely driven by scaling language models to unprecedented sizes. Various large-language model (LLM) scaling laws have been formulated (Sardana et al., 2024), with perhaps the most influential being the Chinchilla law, which demonstrates that parameters and tokens scale approximately linearly as the model scales (Hoffmann et al., 2024). Many subsequent LLMs have been trained following this model (Rae et al., 2021), with some models including the Llama 2 and Llama 3 family of models being trained on 2 and 15 trillion tokens respectively, far more than the 'optimal' amount according to the Chinchilla scaling law (Dubey et al., 2024). However, it is often prohibitive to train such large models, and impractical to continue scaling with the amounts of data required to train such models.

This has sparked interest in small language models (Schick and Schütze, 2021; Magister et al., 2023) with much fewer parameters, requiring much less data for training. While much research has been conducted on knowledge distillation and improving the model architecture for small language models, comparably less research has investigated the contributions of different types of data used for model training, which is arguably just as important. Indeed, because LLM pretraining data typically comprises a mix of sources (Chowdhery et al., 2023), researchers have found that the composition of pretrained data greatly affects model performance (Du et al., 2022; Wei et al., 2015), though determining the optimal recipe for pretraining data is challenging. Recent research exploring optimization of pretraining data for LLMs at scale includes DoReMi, which trains a small proxy model to produce domain weights for downstream tasks, and then uses the model to resample the dataset for training huge LLMs (Xie et al., 2024). However, the question of how to choose data for sample-efficient training of small language models, such as in cases where computational resources are limited, has received little attention.

Psycholinguistic precedent exists for sample-efficient pretraining; children see much less words than a modern LLM yet perform exceptionally well on reasoning tasks. For example, Chinchilla sees over 10000 times the number of words a 13 year old child has ever encountered (Choshen et al., 2024). By the time typical English-speaking children at around 6 years old have obtained adult-level grammatical knowledge (Kemp et al., 2005), they have seen only around 10-50M words (Hart et al., 1997; Huebner et al., 2021). In comparison, Llama-3 is

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 284–291
November 15-16, 2024

trained on 15T tokens (Dubey et al., 2024). Given the great disparity between the amount of training data an LLM requires and what children require, it seems worthwhile to investigate whether training LLMs can be as sample efficient.

BabyBERTa (Huebner et al., 2021) attempts to address this, showing that when training a model on data similar to what is seen by children between the ages 1 and 6, it is able to acquire grammatical knowledge similar to pretrained RoBERTa-base, but with around 15X fewer parameters and 6,000X fewer words; this indicates that utilizing child-directed input may be advantageous for more sample efficient pretraining (Huebner et al., 2021). Similarly, Eldan and Li (2023) follow suit, releasing TinyStories, a synthetic dataset of short stories that only contain words that typical 3- to 4-year-old children understand. They demonstrate that TinyStories can be leveraged to train language models with much less parameters than SOTA models, yet still produce coherent output with almost perfect grammar as well as emergent reasoning abilities. Along the same vein, GPT-wee (Bunzeck and Zarrieß, 2023) shows that child-directed speech can be used with curriculum learning for simulating children's learning as a potential solution to sample-constrained training.

In this paper, we evaluate the effect of different datasets on model performance for sample efficient model training. In our case, we limit our training dataset to 10M words, in accordance with the BabyLM Challenge's super-strict track (Choshen et al., 2024). We consider several different types of datasets, namely child-directed speech (CHILDES), classic books (Gutenberg), a mixed dataset (Mix) and the TinyStories dataset. Experimental results show that smaller models benefit from training on diverse datasets like Mix on the BabyLM evaluation suite (Choshen et al., 2024), but larger models perform better when trained on more complex and rich datasets like Gutenberg. Our findings suggest that the optimal dataset depends on the model size and that neither child-directed speech nor child-directed stories are optimal for language models of any sizes.

2 Dataset

For our experiments, we obtained datasets from the BabyLM Challenge (Choshen et al., 2024). Individual categories of 10M-word datasets were procured by extracting the first 10M words from that category in the 100M-word dataset of the BabyLM challenge. We also used Mix, the 10M-word developmentally-plausible corpus of BabyLM, and TinyStories. To measure for complexity in the language of these datasets, we use several readability metrics, including the Flesch reading ease (FRE) score (Flesch, 1948), ARI (Automated Readability Index) (Smith and Senter, 1967), and the Gunning fog index (Gunning, 1969).

For a document $d_i \in \mathcal{C}$, its FRE score is computed as:

$$\text{FRE}(d_i) = 206.835 - (1.015 \cdot \text{ASL}) - (84.6 \cdot \text{ASW})$$

where ASL is the average sentence length (the number of words divided by the number of sentences) and ASW is the average number of syllables per word (the number of syllables divided by the number of words). Higher FRE scores correspond to simpler texts (e.g., children's literature), while lower scores indicate more complex writing (e.g., machine learning papers). The ARI score is calculated as:

$$\text{ARI}(d_i) = 4.71 \cdot \left(\frac{\text{characters}}{\text{words}}\right) + 0.5 \cdot \left(\frac{\text{words}}{\text{sentences}}\right) - 21.43$$

Higher ARI scores indicate more complex text requiring higher grade levels to comprehend. The Gunning fog index score is calculated as:

$$\text{Fog}(d_i) = 0.4 \cdot \left[\left(\frac{\text{words}}{\text{sentences}}\right) + 100 \cdot \left(\frac{\text{complex words}}{\text{words}}\right)\right]$$

Like ARI, higher Gunning fog scores indicate more complex text.

Our individual datasets comprise:

- **CHILDES**: The CHILDES dataset is composed of examples of the human language acquisition process starting from a very young age (MacWhinney, 2000). We constructed a 10 million word training corpus from the CHILDES portion of the small track (100M). We took the first 10M words from the CHILDES portion.

- **Gutenberg**: The Gutenberg dataset is a large dataset composed of English language books (Gerlach and Font-Clos, 2020). We took the first 10M words from the Gutenberg portion of the small track dataset.

- **Mix (Default)**: This was the default 10M dataset for the strict-small track. The split of is displayed below:

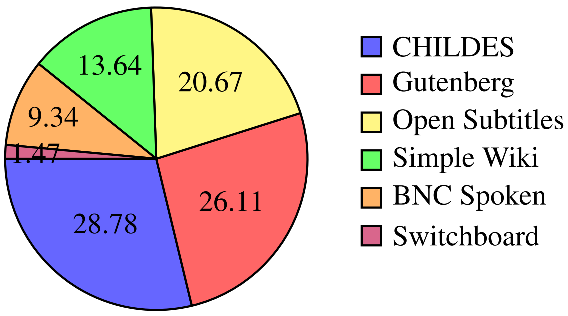


Figure 1: Default dataset composition

- **TinyStories**: We took the first 10M words from the TinyStories dataset on Hugging Face[1] (Eldan and Li, 2023). (FRE = 105.19)

Dataset	FRE	Gunning Fog	ARI
Mix	105.89	5.62	1.59
CHILDES	115.70	2.84	0.20
Gutenberg	87.49	9.89	7.12
TinyStories	105.19	4.83	0.85

Table 1: Readability metrics across different datasets. Lower FRE and higher Gunning Fog and ARI scores indicate a more complex dataset.

3 Methodology

3.1 Preprocessing

For both pre-processing and model training, we built off the BabyLlama repository[2] (Timiryasov and Tastet, 2023). Following their pre-processing steps, we applied regex-based cleaning and trained a Byte-Pair Encoding tokenizer on the training sets of whatever dataset we were working with. The train and dev sets were split into 128-token chunks, with the model being presented a new random permutation of these chunks in each epoch. Validation loss is computed at the end of each epoch using a fixed, randomly sampled subset of the dev set.

3.2 Training

Given that this builds upon the TinyStories paper, focused on dataset optimization for very small language models, we focused mainly on GPT models of sizes 18M, 44M and 97M, which we trained on various datasets. We used this to explore whether different model sizes would affect which dataset performed the best. We trained for 4 epochs, using consistent hyper-parameters. Subsequently, we trained a Llama-20M model to confirm that the same pattern regarding dataset complexity is observed in Llama models as well. Lastly, large model baselines of GPT2-705M and Llama-360M are used, as these were the original parent model sizes originally used by last year's BabyLM winning model (Timiryasov and Tastet, 2023).

3.3 Evaluation

Evaluation of model performance was done using the BabyLM evaluation suite (Choshen et al., 2024). This consists of the following benchmarks:

- **BLiMP:** BLiMP (Benchmark of Linguistic Minimal Pairs for English) evaluates language models on their ability to identify grammatical acceptability. It presents pairs of sentences that differ by one linguistic element, testing the model's understanding of 12 areas of English morphology, syntax, and semantics, such as anaphor agreement and filler-gap constructions. It measures how well models assign higher probability to the grammatically correct sentence in each pair. (Warstadt et al., 2020)

- **EWoK:** EWoK (Elements of World Knowledge) evaluates language models on their ability to build and apply internal world models. It tests models' understanding of concepts and contexts by presenting them with minimal pairs of scenarios where the models determine the plausibility of context and target combinations. (Ivanova et al., 2024)

- **GLUE:** GLUE (General Language Understanding Evaluation) evaluates language models on a variety of natural language understanding tasks. It covers tasks such as sentiment analysis, text similarity, question answering, and textual entailment. (Wang et al., 2018) Unlike in the BabyLM evaluation suite, however, we do not do finetuning in this case and run it as a zero-shot evaluation due to computational constraints.

4 Results and Discussion

Overall, our results demonstrate that the effectiveness of a training dataset is dependent on the model size. Specifically, smaller models (with fewer parameters) benefit more from training on a diverse

[1] https://huggingface.co/datasets/roneneldan/TinyStories
[2] https://github.com/timinar/BabyLlama

Model	Dataset	BLiMP Supplement	BLiMP Filtered	EWoK	Macroaverage
GPT2-18M	CHILDES	52.8	58.2	50.5	53.83
	Gutenberg	55.7	62.4	50.3	56.13
	Mix	**55.9**	**63.7**	49.7	**56.43**
	TinyStories	55.2	57.5	**50.7**	54.47
GPT2-44M	CHILDES	55.3	57.8	**51.2**	54.77
	Gutenberg	57.6	63.0	50.0	56.87
	Mix	**58.2**	**65.6**	50.4	**58.07**
	TinyStories	52.8	57.1	50.4	53.43
GPT2-97M	CHILDES	49.7	60.5	49.6	53.27
	Gutenberg	**59.0**	**65.3**	**51.1**	**58.47**
	Mix	58.0	66.0	50.6	58.20
	TinyStories	54.6	59.1	50.3	54.67
Llama-20M	CHILDES	53.4	57.9	50.2	53.83
	Gutenberg	**57.4**	60.0	**50.6**	**56.00**
	Mix	56.6	**62.8**	50.2	56.53
	TinyStories	46.7	51.1	49.8	49.20
GPT2-705M	Gutenberg	**59.9**	**66.8**	**50.6**	**59.10**
	Mix	56.7	66.1	50.6	57.80
Llama-360M	Gutenberg	**56.7**	**66.5**	50.2	**57.80**
	Mix	56.6	62.8	**50.5**	56.63

Table 2: Summary of BLiMP filtered, BLiMP supplement, EWoK results, and Macroaverage for various models and datasets

dataset like Mix, while larger models show improved performance when trained on the Gutenberg dataset. As shown in Table 2, for smaller models like GPT2-18M and GPT2-44M, Mix consistently achieves the best performance on BLiMP, scoring 63.7 and 65.6 respectively on BLiMP Filtered, and 55.9 and 58.2 on BLiMP Supplement. However, as we move to larger models like GPT2-97M and GPT2-705M, the Gutenberg dataset takes the lead, achieving the highest scores across most metrics (59.0 and 59.9 on BLiMP Supplement, 65.3 and 66.8 on BLiMP Filtered). We see this also extend to the Llama models as well, where the larger Llama-360M performs best with Gutenberg data (56.7 on BLiMP Supplement and 66.5 on BLiMP Filtered), while the smaller Llama-20M shows mixed results between Gutenberg and Mix. Interesting, both CHILDES and TinyStories consistently underperform across all model sizes, with scores typically lower than both Mix and Gutenberg datasets. On the other hand, we see a very different story when looking at macro average GLUE scores for the models (Table 3), with TinyStories performing well for small models and CHILDES performing well for the big model. However, when examining the GLUE subtasks further, we do not see a clear trend on which dataset type results a stronger performance, and cannot conclude a clear trend here.

4.1 Dataset and model performance

Model performance results on various datasets was observed in table 2. Small models, such as GPT2-18M and GPT2-44M, have limited capacity due to fewer parameters. This constraint affects their ability to capture complex linguistic patterns and nuanced language structures. Datasets like Gutenberg with a relatively lower FRE score (87.49) contain wider vocabulary, more intricate syntax, and nuanced semantic meaning. Due to their limited capacity, small models cannot fully learn from the complexity of the dataset. They oversimplify the language patterns, leading to high bias and poor generalization. This underfitting results in lower performance on evaluation benchmarks.

In contrast, larger models, such as GPT2-97M, GPT2-705M, and LLaMA-360M, possess greater capacity to learn and represent complex patterns due to their increased number of parameters. Because the Gutenberg dataset, consisting of a diver-

Model	Dataset	MRPC	MultiRC	QNLI	SST-2	BoolQ	MNLI	QQP	WSC	RTE	Cola (MCC)	Macro Average
GPT2-18M	CHILDES	34.31	45.50	50.92	53.67	58.59	32.42	42.47	38.46	48.20	-0.07	40.45
	Gutenberg	35.78	52.56	49.52	47.94	46.73	32.74	60.68	61.54	53.24	0.05	44.08
	Mix	58.33	44.35	47.14	47.71	57.00	32.42	46.77	46.15	44.60	0.03	42.45
	TinyStories	60.78	42.86	51.72	51.83	62.63	32.56	50.54	42.31	48.20	0.06	44.35
GPT2-44M	CHILDES	46.57	42.41	51.13	47.71	55.96	32.84	41.52	53.85	46.76	0.07	41.88
	Gutenberg	64.71	45.54	50.88	50.92	60.98	31.85	37.32	38.46	43.88	-0.02	42.45
	Mix	52.94	47.07	50.62	48.17	55.23	32.42	54.06	38.46	42.45	0.03	42.14
	TinyStories	45.59	53.09	47.04	48.39	42.26	33.19	62.01	59.62	54.68	-0.06	44.58
GPT2-97M	CHILDES	57.35	53.42	49.27	50.23	44.59	35.76	62.47	61.54	53.96	0.06	46.86
	Gutenberg	54.90	47.69	50.62	53.21	54.98	31.46	38.67	38.46	43.17	0.03	41.32
	Mix	47.05	49.88	48.57	50.00	44.10	33.48	61.82	61.54	56.12	-0.05	45.25
	TinyStories	65.20	43.61	50.40	51.83	62.08	32.03	38.05	44.23	50.36	0.07	43.79

Table 3: Detailed GLUE scores for various GPT models and datasets

sity of subject materials (Gerlach and Font-Clos, 2020), offers the most nuanced sentence structures and vocabulary out of all the datasets, it could be argued that diversity within the dataset may be more important than having a diverse basket of datasets for models with a higher number of parameters.

4.2 Dataset Convergence

In our experiments, CHILDES converged faster than either then Gutenberg or the Mix datasets for both GPT2-44M and GPT2-18M models. This can be observed in figure 2 and 3 below, and can be explained by the nature of CHILDES dataset. The higher FRE score (115.70) of this child-directed speech dataset indicates simpler grammatical structures, shorter sentences, and straightforward syntax compared to the adult-oriented language found in datasets like Gutenberg or Mix. In addition, because caregivers frequently repeat words and phrases when interacting with children, the dataset is characterized by high repetition, making the learning task of capturing the underlying structures and relationships in the data easier and faster to converge quickly during training. In short, due to the low perplexity of the CHILDES dataset, the model has less uncertainty in predicting the next word in a sequence, resulting in a smoother loss landscape and simplifying the learning task.

4.3 Underperformance of Child-directed and Synthetic Datasets

Neither the CHILDES nor TinyStories datasets performed very well on the BLiMP or EWoK evaluation suite (Choshen et al., 2024). The CHILDES dataset consistently underperformed no matter the model size, suggesting that child-directed speech may not be not advantageous for training a robust model. This is consistent with the lack of success

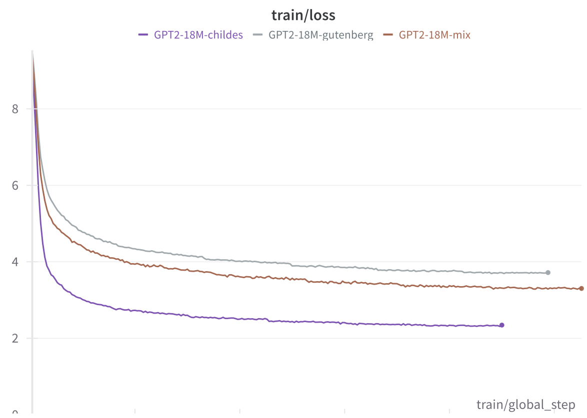


Figure 2: Train loss when training GPT2-18M on various datasets

in implementing curriculum learning for child data in the previous BabyLM challenge (Bunzeck and Zarrieß, 2023). In their paper, Bunzeck and Zarrieß noted that the integration of more sophisticated linguistic factors into the training process might be needed, as their curriculum approach based on prototypicality measures didn't effectively capture the language acquisition process they were looking for.

Considering the strong performance of TinyStories in (Eldan and Li, 2023), and the fact that we adopted the same GPT-44M architecture as in paper, with a hidden size of 768, 2 layers and 8 heads, we were surprised by the poor performance of the TinyStories dataset. That said, we only used a 10M subset of TinyStories, and given its limited vocabulary and grammatical range (and higher FRE score of 105.19), perhaps there was insufficient diversity and exposure to new formats as previously discussed. Additionally, we utilized different benchmarks. The BLiMP and EWoK benchmarks assess a model's understanding of complex grammatical rules and world knowledge; this is not likely to be adequately covered by the TinyStories dataset. In

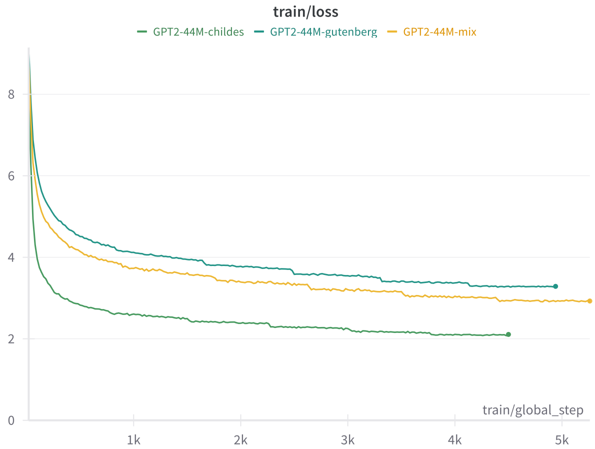


Figure 3: Train loss when training GPT2-44M on various datasets

short, models trained on TinyStories may lack exposure to the types of linguistic phenomena these benchmarks evaluate.

The disparity in TinyStories' performance across benchmarks likely stems from the divergent linguistic and cognitive demands of each dataset. GLUE evaluates general-purpose natural language understanding (NLU) tasks, such as sentiment analysis and paraphrase identification, which align well with the broad, semantic patterns learned from narrative content in TinyStories. In contrast, BLiMP emphasizes fine-grained syntactic and grammatical competence, while EWoK assesses factual reasoning and contextual world knowledge—skills that TinyStories' simplified narrative structure and limited syntactic diversity do not comprehensively support. Consequently, while TinyStories provides effective training for NLU, it lacks the complexity required for the precise linguistic and knowledge-based reasoning assessed by BLiMP and EWoK.

On the whole, however, we do not see the huge performance gains that were reported in the original TinyStories paper. The success of TinyStories in the original paper may perhaps be partially attributed to the narrative structure of the data, which provides contextual coherence and sequential dependencies that models can leverage. However, given that the Gutenberg dataset also contains narrative texts but with more complicated language and storylines, it offers better training data for models to learn general language patterns.

5 Limitations

Our study has several limitations. First, we used consistent hyper-parameters across all experiments for comparability, but this may not have been optimal for each model-dataset pair. Tuning hyper-parameters individually could have yielded better performance.

Second, the BLiMP and EWoK benchmark assess linguistic competence on tasks on represented in datasets such as TinyStories or CHILDES, potentially biasing the evaluation. In short, there is a mismatch between the training data afforded by child datasets and the test set.

Lastly, due to computational limitations, models were trained for only four epochs. Longer training might have allowed models to better capture the nuances of the datasets.

6 Conclusion and Future Work

In this paper, we investigated the impact of dataset composition on the performance of small language models in a sample-efficient training regime. By training models of varying sizes on different datasets limited to 10 million words, we sought to identify which types of data are most beneficial for language acquisition in resource-constrained settings.

We found that tiny models (e.g., GPT2-18M and GPT2-44M) performed best when trained on the Mix dataset, which offers a diverse combination of language inputs, while slightly larger small language models achieved superior performance when trained on the Gutenberg dataset, leveraging its linguistic richness. In contrast, models trained on CHILDES or TinyStories underperformed regardless of size.

For future work, a more thorough investigation of other types of data sources such as news articles, scientific texts, and conversational data might better tease out the optimal dataset for model performance. Additionally, it might be useful to explore curriculum learning, which presumable models the developmental process of a language learning child.

Widening the benchmarks beyond GLUE and BLiMP tasks to coherent text generation, as well as scaling dataset sizes and tasks would allow for a more comprehensive and robust study as well.

Acknowledgments

We thank Stanford University for their support for this paper.

References

Bastian Bunzeck and Sina Zarrieß. 2023. Gpt-wee: How small can a small language model really get? In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 35–46.

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*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

Rudolf Franz Flesch. 1948. A new readability yardstick. *The Journal of applied psychology*, 32 3:221–33.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1).

Robert Gunning. 1969. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13.

Betty Hart, Todd R Risley, and John R Kirby. 1997. Meaningful differences in the everyday experience of young american children. *Canadian Journal of Education*, 22(3):323.

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, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. 2024. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

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

Nenagh Kemp, Elena Lieven, and Michael Tomasello. 2005. Young children's knowledge of the" determiner" and" adjective" categories.

Brian MacWhinney. 2000. *The Childes Project: Tools for Analyzing Talk, Volume II: the Database*, 3rd edition. Psychology Press.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, Toronto, Canada. Association for Computational Linguistics.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John F. J. Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, L. Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, N. K. Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Tobias Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew G. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem W. Ayoub, Jeff Stanway, L. L. Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021.

Scaling language models: Methods, analysis & insights from training gopher. *ArXiv*, abs/2112.11446.

Nikhil Sardana, Jacob Portes, Sasha Doubov, and Jonathan Frankle. 2024. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. *Preprint*, arXiv:2401.00448.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

E.A. Smith and R.J. Senter. 1967. *Automated Readability Index*. AMRL-TR. Aerospace Medical Research Laboratories, Aerospace Medical Division, Air Force Systems Command.

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. *Preprint*, arXiv:2308.02019.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. 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.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. 2015. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2024. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36.

BabyLlama-2: Ensemble-Distilled Models Consistently Outperform Teachers With Limited Data

Jean-Loup Tastet
University of Copenhagen
Department of Computer Science
Copenhagen, Denmark
jeta@di.ku.dk

Inar Timiryasov
University of Copenhagen
Niels Bohr Institute
Copenhagen, Denmark
inar.timiryasov@nbi.ku.dk

Abstract

We present BabyLlama-2, a 345 million parameter model distillation-pretrained from two teachers on a 10 million word corpus for the BabyLM competition. On the BLiMP and SuperGLUE benchmarks, BabyLlama-2 outperforms baselines trained on both 10 and 100 million word datasets with the same data mix, as well as its teacher models. Through an extensive hyperparameter sweep, we demonstrate that the advantages of distillation cannot be attributed to suboptimal hyperparameter selection of the teachers. Our findings underscore the need for further investigation into distillation techniques, particularly in data-limited settings.

1 Introduction

With frontier model training runs using beyond 10^{25} FLOPs (Dubey et al., 2024), training efficiency has become a billion-dollar question. Humans are vastly more sample efficient than current Large Language Models (LLMs). For example, a typical 13-year-old child has been exposed to less than 100 million words (extrapolating from Gilkerson et al. (2017)), whereas Llama-3.1 has been trained on 15.6 trillion text tokens. The goal of the BabyLM Challenge (Choshen et al., 2024) is to optimize pretraining given dataset limitations inspired by human development.

In this work, we present our contribution to the BabyLM challenge (Strict-Small Track), with the following main results:

- BabyLlama-2 model: This 345M parameter decoder-only model[1], distillation-pretrained on 9.5M words, outperforms baseline models trained on both 10M and 100M words (using the same data mix). It also surpasses similar models pretrained using conventional methods.
- Extensive hyperparameter sweep: We have conducted a comprehensive hyperparameter optimization and demonstrated that distillation pretraining consistently outperforms the best models from the sweep.
- Correlation between test loss and performance: As a byproduct of our sweep, we have identified a correlation between zero-shot performance on the BLiMP task and the model's test loss.

The success of distillation pretraining, i.e. pretraining from scratch with distillation loss, in our experiments highlights its potential as a powerful technique for improving model performance, especially in data-limited settings. While our findings are promising, they also raise intriguing questions about the nature of knowledge distillation and its interaction with pretraining objectives. Further investigation into these areas could yield valuable insights for the development of more sample-efficient language models.

2 Related Work

The first edition of the BabyLM challenge, which aims to optimize language model pretraining under data constraints inspired by human language acquisition, prompted numerous works on sample-efficient pretraining. For a detailed summary of all contributions, see the review by Warstadt et al. (2023).

Outside the BabyLM context, relatively few works address training on limited language datasets. Notable exceptions include Muennighoff et al. (2023), who studied the scaling of data-constrained

[1] It is worth noting that encoder models are better suited for the evaluation tasks of the challenge than decoder ones. In last year's evaluation (Warstadt et al., 2023), the 125M parameter RoBERTa-base (Liu et al., 2019) performed on par with the 70B parameter Llama-2 (Touvron et al., 2023b). However, our focus throughout this paper shall be on generative, decoder models.

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 292–301
November 15-16, 2024 ©2024 Association for Computational Linguistics

LLMs. Their main finding is that training for more than 4 epochs leads to diminishing returns. Luukkonen et al. (2023) trained FinGPT on more than 30B tokens in Finnish language. Although resource-constrained, this dataset is significantly larger than that of the BabyLM Challenge. A sample-efficient modification of BERT architecture was proposed by Samuel et al. (2023), with a model trained on a 100M word dataset from the British National Corpus outperforming the original BERT model.

The existing literature on training small models often focuses on models deployable on edge devices, such as MobileLLM (Liu et al., 2024). However, these works typically concentrate on deployment efficiency rather than sample efficiency.

Knowledge distillation has recently attracted significant attention, primarily for deployment efficiency reasons (see Xu et al. (2024) for a systematic review). Typically, this involves using large frontier models as teachers to train smaller student models. In contrast, BabyLlama-2 utilizes distillation for sample-efficient pretraining, using similar-sized teacher models trained on the same limited dataset.

A similar phenomenon, where a student model outperforms its teachers when distilled from models with identical architecture and trained on the same dataset, was observed in "Born-Again Neural Networks" (Furlanello et al., 2018). However, this work did not focus on the data-limited regime and it used LSTM variants (instead of transformers) for language modeling.

3 Background

Knowledge distillation, introduced by Hinton et al. (2015), is a technique for transferring knowledge from a "teacher" model to a "student" model. The core idea is to train the student to mimic the logit distribution (soft targets) produced by the teacher, rather than just the hard labels of the training data. The distillation loss combines the standard cross-entropy loss with the soft target loss:

$$\begin{aligned}\mathcal{L}_{\text{distill}}(y, z_s, z_t) = \alpha\, \mathcal{L}_{\text{CE}}(y, \sigma(z_s)) + \\ (1-\alpha)\, T^2\, D_{\text{KL}}\left(\sigma(z_t/T)\,||\,\sigma(z_s/T)\right)\end{aligned} \quad (1)$$

where α balances the usual cross-entropy loss $\mathcal{L}_{\text{CE}}$ and the soft targets loss, T is the temperature parameter that softens the probability distributions, z_s and z_t are respectively the logits of the student and teacher models, σ is the softmax function, and D_{KL} denotes the Kullback-Leibler divergence. In our implementation, we use the averaged logits of an ensemble of teacher models as z_t. Moreover, unlike typical applications, our student and teacher models are of the same size.

4 Model

Architecture. Previous experiments have shown that the Llama architecture (Touvron et al., 2023a), featuring RoPE and a SwiGLU non-linearity, requires fewer epochs to reach minimal loss compared to GPT-2 or GPT-J architectures (Timiryasov, 2023). After training a family of Llama models ranging from 16M to 728M parameters, we converged on a specific 345M model architecture suggested in MobileLLM (Liu et al., 2024) and also used in SmolLM (Allal et al., 2023), whose hyperparameters are listed in table 1. This design incorporates Grouped-Query Attention (GQA) and prioritizes depth over width. Some details of our model selection are listed in appendix B.

Hyperparameter	Value
Vocabulary size	16,000
Number of layers	32
Number of heads	15
Number of KV heads	5
Embedding dimension	960
Hidden dimension	2560
Total parameters	345M

Table 1: BabyLlama-2 Model Architecture.

Pretraining Approach. The particularity of the BabyLlama-2 model is to be distilled from an ensemble of teacher models, using the distillation loss (1). The teacher models share the same architecture and are pretrained on the same dataset using the standard cross-entropy loss. The student model is then pretrained with the same hyperparameters, using the mean teacher logits $\bar{z}_t$ in the distillation loss $\mathcal{L}_{\text{distill}}(y, z_s, \bar{z}_t)$.

5 Experimental Setup

Dataset. We use the 10 million word BabyLM-2 dataset (Zhuang et al., 2024), that we split into 9.5M train and 0.5M validation splits, as well as the accompanying 10M word "dev" dataset, that we use as a test split. While the validation split is used to perform the hyperparameter optimization,[2]

[2]This choice is dictated by the following logic. A hyperparameter sweep can be viewed as a form of optimization. There-

the test split is used solely for the purpose of reporting the final cross-entropy loss. Each dataset is composed of six files, corresponding each to a different type of (English) language that a child is likely to be exposed to, such as transcribed child-directed speech, children's books, subtitles, or simple Wikipedia. The relative fractions of these files differ slightly between, on the one hand, the train and validation splits and, on the other, the test split, which is therefore slightly out of distribution.

We have experimented with the FineWeb-Edu dataset (Lozhkov et al., 2024) but have observed that models trained on the BabyLM-2 dataset reach better BLiMP scores (see appendix C for more details).

Training. The teacher models are pretrained using the `Trainer` class from the HuggingFace Transformers library, using the hyperparameters listed in table 2. For the distillation, we use the modified trainer from the original BabyLlama (Timiryasov and Tastet, 2023b), with one, two or three teachers. We use the AdamW optimizer (Loshchilov and Hutter, 2019), with a cosine schedule for the learning rate and 600 warm-up steps. The pretraining hyperparameters have been optimized using a coarse-grained scan, with each parameter being varied independently. The distillation hyperparameters α and T were optimized similarly, while holding the pretraining parameters fixed.

All models share the same Byte-Pair Encoding (BPE) tokenizer with a vocabulary size of 16000 trained on the training split of BabyLM-2 dataset.

Hyperparameter	Value
Learning rate	$7 \cdot 10^{-4}$
Number of epochs	8
Batch size	128
Weight decay	5
Distillation T	1
Distillation α	0.5

Table 2: Training and distillation hyperparameters of BabyLlama-2.

Hyperparameter Sweep. To exclude the possibility that the student model BabyLlama-2 outperforms its teachers due to a suboptimal choice of hyperparameters for the teachers, we have performed a comprehensive sweep for the teachers' hyperparameters using the W&B API (Biewald, 2020). We vary the following hyperparameters: the learning rate and its schedule, the Adam parameters (β_1, β_2, ϵ), the batch size, the number of epochs and warm-up steps, the weight decay, the maximum gradient norm, and the attention dropout. We use the Bayesian Optimization and Hyperband (BOHB) (Falkner et al., 2018) parallel sweep algorithm, which stops badly-performing runs early, and we minimize the validation loss at the last epoch. Suitable priors are used for each parameters, usually log-normal or log-uniformly distributed around the values obtained from the coarse-grained scan, with the exception of the attention dropout (uniform) and schedule (discrete). In total, we trained 265 models as part of the sweep, amounting to 26 GPU-days. While the sweep produced some runs that perform noticeable better than the teachers trained with the parameters in table 2, re-training them from a different initial state, but otherwise with the exact same parameters, lead to models that significantly under-performed compared to the initial teachers. Due to this lack of stability with respect to the initialization, we decided to use the original teachers for the distillation procedure.

Benchmarks. We evaluate the performance of the teacher and student models on the benchmarks suggested by the organizers of the BabyLM challenge. Those include zero-shot benchmarks — such as BLiMP (Warstadt et al., 2020), which focuses on linguistic knowledge in English, and EWoK (Ivanova et al., 2024), focusing on world knowledge — as well as the suite of fine-tuning benchmarks SuperGLUE (Wang et al., 2020) about language understanding. For the latter, the fine-tuning hyperparameters are optimized using a separate sweep for each task (totalling 1293 runs and 37 GPU-days). The optimal parameters, listed in table 4, differ significantly from the suggested defaults. See appendix A for further discussion.

Baseline models. The organizers of BabyLM-2 have provided two baseline models: LTG-BERT (Samuel et al., 2023), (encoder-only) and BabyLlama (Timiryasov and Tastet, 2023a) (decoder). Both models were re-trained by the challenge organizers on both the 10M and 100M word datasets. LTG-BERT modifies the original BERT architecture by utilizing the pre-norm variant of the

fore we would consider using the dev split from BabyLM-2 as a violation of the rules of the challenge. Of course, it means that we trained only on 95% of the tokens, and could potentially improve our results further.

transformer with GEGLU feed-forward layers and by disentangling positional information from token embeddings. The highest performing solution of the 2023 edition of the BabyLM challenge, ELC-BERT (Charpentier and Samuel, 2023), is based on this architecture. On the other hand, BabyLlama (the highest-performing decoder model) uses the standard LLaMA architecture (Touvron et al., 2023a), but a modified training procedure, following a similar approach to the one presented here. However, in contrast to BabyLlama-2, it was distilled from two larger teachers with two different architectures (GPT and Llama), and had six times less parameters. Since BabyLlama-2 aims to demonstrate the validity of the ensemble distillation method itself, it uses same-size, homogeneous models in order to remove potential confounding factors. In addition to the baseline models, we vary the number of teachers between 1 and 3, and compare BabyLlama-2 to the ensemble formed by the two teacher models (applying softmax to the averaged logits $\bar{z}_t$ and letting the gradient flow back into both teachers during fine-tuning, with the same training hyperparameters as for BabyLlama-2). When evaluating the original BabyLlama on the SuperGLUE benchmarks, we fine-tune it again using the hyperparameters reported in (Timiryasov and Tastet, 2023a), and successfully reproduce all of its scores.

6 Results

Figure 1 summarizes the performance of the models considered in section 5 with respect to the various evaluation metrics: the cross-entropy loss evaluated on the held-out test set, the BLiMP scores for the "filtered" and "supplement" subsets of evaluation tasks, and the mean SuperGLUE score. The EWoK benchmark is not shown, since the performance of our models and of the baselines trained on 10M words is consistent with random chance, hinting that all these models have extremely limited world knowledge, if any.

Distributions. Violin plots are used in order to quantify the variability across model initializations, with a minimum of 5 runs per model. Each subplot shows a different metric, with the y-axis listing the various models considered: the teacher models, pretrained without distillation; the student models pretrained with one, two or three teachers; the direct ensembles formed by averaging the logits of two teachers; the baseline models for the 2024 BabyLM challenge; and the 265 models from the hyperparameter sweep. No violin is shown for baseline models, since they do not have an associated distribution. Similarly, running fine-tuning benchmarks for all the models from the sweep would have been computationally prohibitive, therefore the SuperGLUE distribution associated with the sweep is not present, with only the best checkpoint being shown.

Models of interest. Instead of, or in addition to the distributions, the performance of various models of interest is plotted using markers. This includes the baseline models, denoted by triangles for BabyLlama and squares for LTG-BERT, with filled markers for baselines pretrained on the 10M word dataset and empty markers for the 100M one. We also indicate with stars the two BabyLlama-2 models that have been submitted to the 2024 edition of the BabyLM challenge. Finally, the cross denotes the best model from the entire sweep, as quantified by its validation loss. The detailed numerical results for the models of interest are listed in table 3, and table 5 further details the SuperGLUE scores of the two submitted BabyLlama-2 checkpoints.

Cross-entropy. The cross-entropy loss is by far the cleanest metric, with a standard deviation across initializations much smaller than the difference between models.[3] It shows a clear and gradual improvement between the teacher models, the student models trained from a single teacher, those trained from two teachers, and those trained from three teachers, although we note that there are diminishing returns as we add more teachers. Even with a single teacher, the improvement is larger that what can be achieved through the hyperparameter sweep. However, looking at the BabyLlama baseline[4], it is clear that this improvement is nowhere near the one resulting from using a ten-fold larger dataset. The cross-entropy loss of the direct ensemble of two teachers is almost as low as for the corresponding model obtained through distillation.

[3]The much larger standard deviation for the sweep comes from including all runs (including early and badly performing runs) instead of just the best runs. The relevant quantity for the sweep is therefore the edge of the distribution. The "best" model is not always located on this edge, since the validation loss does not correlate perfectly with the test loss or the benchmark scores.

[4]The cross-entropy loss is not shown for the LTG-BERT baseline, since it is an encoder-only model trained using masked language modeling, and as such its loss is not comparable to the one discussed here.

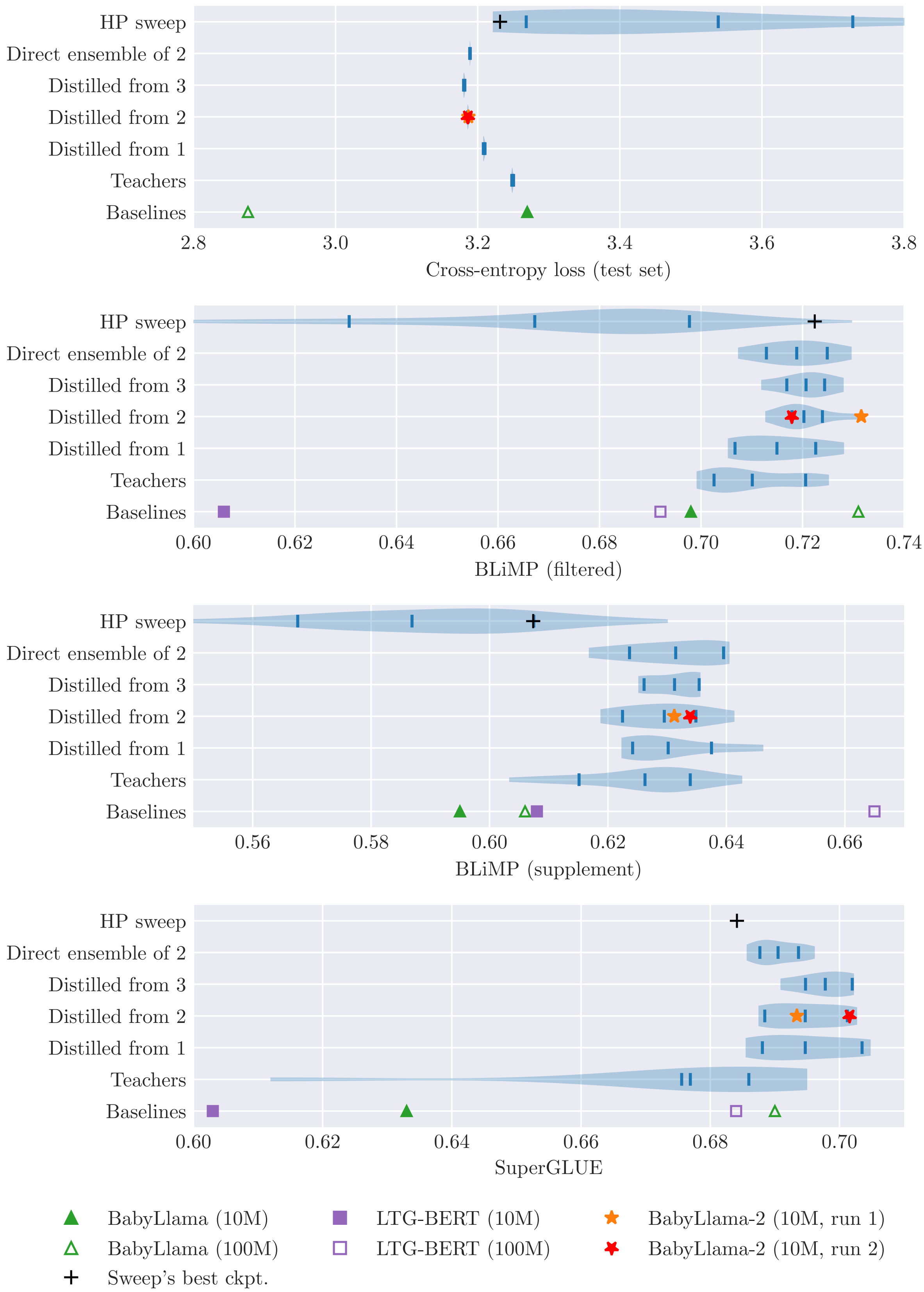


Figure 1: Comparison of the models for each evaluation metric, in the form of violin plots, with ticks denoting the mean and ± 1 standard deviation. The baselines are denoted by square and triangle markers, the submitted model (BabyLlama-2) by stars, and the best checkpoint from the entire hyperparameter sweep by a cross. BabyLlama (100M) and LTG-BERT (100M) were trained on the 100M dataset.

Model	BLiMP (filtered)	BLiMP (supplement)	EWoK	SuperGLUE	Macro-average
BabyLlama-2 (run 1)	**73.2**	63.1	50.6	69.3	64.0
Teacher 1	71.9	61.8	50.6	61.2	61.3
Teacher 2	72.1	62.9	50.1	69.5	63.6
BabyLlama-2 (run 2)	71.8	**63.4**	**51.5**	**70.2**	**64.2**
Teacher 1	70.9	62.9	50.4	67.6	62.9
Teacher 2	70.5	62.4	51.1	68.4	63.1
Sweep's best ckpt.	72.2	60.7	50.1	68.4	62.9
BabyLlama (10M)	69.8	59.5	50.7	63.3	60.8
LTG-BERT (10M)	60.6	60.8	48.9	60.3	57.7
BabyLlama (100M)	73.1	60.6	**52.1**	69.0	63.7
LTG-BERT (100M)	69.2	**66.5**	51.9	68.4	64.0

Table 3: Summary of the model scores (in %) across the considered benchmarks. The best scores overall and within the `strict-small` track (10M words maximum) are highlighted.

Benchmarks. The scores on the two BLiMP task sets show a similar trend, but with a significantly higher variability across runs. Because of this, no significant difference is observed between the various distilled or ensemble models. Nonetheless, we can see that the distilled models not only do better than the non-distilled ones, but they tend to achieve this performance more reliably. This is to be contrasted with the performance regression (not shown) that we observed after re-training the best model from the sweep. Direct ensembling leads to similar performance to distillation. Another interesting observation is that despite its much lower cross-entropy loss, the BabyLlama baseline pre-trained on 100M words only performs on par with the best BabyLlama-2 model trained on 10M words on the "filtered" subset of tasks, and significantly underperforms on the "supplement" subset. The results are sensibly similar for the SuperGLUE fine-tuning benchmarks, although with much larger variance among the teacher models. Here, again, the distilled models perform more consistently, and they even beat the two baseline models pretrained on the 100M word dataset. Direct ensembling slightly underperforms compared to distillation, but this could be because fine-tuning introduces a dependence on additional hyperparameters, that have not been precisely re-tuned for direct ensembling.[5]

Relation between loss and benchmark performance. The models trained during the hyperparameter sweep allow us to access the relation between the validation loss and BLiMP scores. First, we observe that the loss on our 0.5M word validation set correlates with the loss on the held-out test set with $R^2 = 0.999$. Second, as can be seen from fig. 2, the validation loss explains a significant portion of the variance of the scores: $R^2 = 0.86$ for for BLiMP Filtered and $R^2 = 0.6$ for BLiMP Supplement.

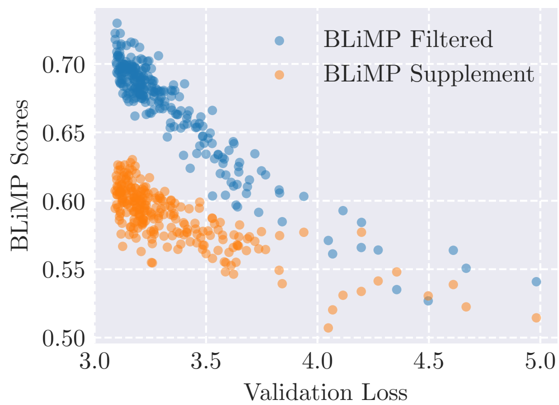


Figure 2: BLiMP scores (averaged over all sub-tasks) as a function of the validation loss. Every circle represents a model from the hyperparameter sweep.

Discussion. The results presented in fig. 1 demonstrate that ensemble distillation from homogeneous teacher models leads to enhanced and more consistent performance across various benchmarks. Notably, BabyLlama-2 often matches or surpasses models pretrained on datasets that are ten times larger. This indicates that the distillation process effectively leverages the knowledge from multiple teachers to compensate for limited data. In addition, the performance of distilled models is consistently as good as, or better than the one of non-distilled models, even when optimizing the hyperparameters of the latter. Therefore, the effect observed in Timiryasov and Tastet (2023a) can-

[5]Naively doubling the fine-tuning learning rate to compensate for the $1/2$ factor resulting from averaging the logits leads to significantly worse performance on SuperGLUE, below that of the teacher models.

not be solely attributed to badly-tuned teacher hyperparameters, and persists even when the student and teachers share the same size and architecture. However, this effect can be difficult to see on the benchmark scores, which are much noisier than the cross-entropy loss. This variability is made particularly evident when looking at the different ordering of the two submitted BabyLlama-2 models across different benchmarks.

Limitations The scalability of the ensemble distillation approach to larger datasets and more substantial model sizes remains unexplored. It is unclear whether the observed benefits will persist or diminish as the scale of data and model parameters increases. Additionally, the exact origin of the improvements from distillation-pretraining remains unclear. Finally, it is not clear whether distillation-pretraining performs significantly better than direct ensembling. Further research, and more sensitive metrics, may be needed to give definitive answers to these question.

7 Conclusions

In this study, we prioritized investigating the robustness of the distillation approach over architectural modifications or dataset curation. Our findings demonstrate that a 345M parameter model, distillation-pretrained on 9.5M words, outperforms models of the same size and architecture pretrained in the usual way. We carried out a systematic analysis to exclude the possibility that the performance gains were due to a single fortunate initialization or suboptimal teacher model hyperparameters. Through an extensive hyperparameter sweep and the training of multiple teacher and student models, we established that distillation-pretraining consistently yields superior performance.

Our results indicate that distillation-pretraining is an effective method for achieving high performance without the need for meticulous hyperparameter tuning, at least within the data-limited regime. The scalability of this approach to larger datasets and model sizes, as well as its applicability to other modalities, remains an open research question.

Acknowledgements

We thank Thea Aarrestad, Erik Dam, Troels C. Petersen, Oleg Ruchayskiy and Raghavendra Selvan for valuable discussions, as well as the anonymous referees for their constructive feedback. This work was supported by a research grant (VIL57416) from VILLUM FONDEN. The work of IT was partially supported by the Carlsberg foundation, and by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No. 847523 'INTERACTIONS'. Computational resources for this work were partially provided by the SCIENCE AI Centre of Copenhagen University.

A SuperGLUE Fine-tuning

The SuperGLUE suite of benchmarks consists of a number of fine-tuning tasks related to language understanding. Since they involve further model training, the scores crucially depend on the chosen fine-tuning hyperparameters. In table 4, we list the hyperparameters used to fine-tune all our models on the SuperGLUE tasks. These parameters were identified using the BabyLlama-2 checkpoint by performing a separate sweep for each task, and then re-starting the fine-tuning with rounded parameters, in order to check the stability of the found parameters. We have observed that they work well with other model checkpoints, including different versions of BabyLlama-2 and teacher models, suggesting that our hyperparameter selection is robust across different model initializations and pretraining objectives (but not model sizes, since the original BabyLlama had different optimal hyperparameters) and is not overfitted to a specific model or task. The detailed SuperGLUE scores of the two BabyLlama-2 checkpoints submitted to the 2024 BabyLM challenge are reported in table 5.

B Scaling Model Size

We performed initial experiments using a small, 16M version of the model, with the same vocabulary size of 16,000; hidden size 256; intermediate size 1024; 8 layers and 8 attention heads. This model can be fully trained in a few minutes but already achieves decent benchmark scores (without distillation, BLiMP Filtered: 0.68, BLiMP Supplement: 0.58).

To understand the relationship between model size and data requirements, we conducted additional experiments with our 16M and 345M models. We trained these models on random (nested) subsets of the 100M word dataset, ranging from 1M to 100M words each (without re-tuning the hyperparameters). Figure 3 illustrates how the loss decreases as the dataset size increases for both the

Task	Max. learning rate	Batch size	Num. epochs	Weight decay	Schedule	Warm-up steps
CoLA	$1 \cdot 10^{-5}$	32	10	0.15	linear	600
SST-2	$2 \cdot 10^{-6}$	24	2	5	constant	200
MRPC	$1 \cdot 10^{-5}$	1	2	2	cosine	500
QQP	$4.5 \cdot 10^{-6}$	32	6	2	linear	500
MNLI(-mm)	$1 \cdot 10^{-5}$	32	2	1	linear	500
QNLI	$5 \cdot 10^{-6}$	32	2	0.3	cosine	200
RTE	$1 \cdot 10^{-5}$	2	2	10	cosine	200
BoolQ	$2 \cdot 10^{-5}$	8	1	0.1	cosine	200
MultiRC	$1 \cdot 10^{-5}$	8	2	2	cosine	500
WSC	$2 \cdot 10^{-6}$	1	24	0.4	cosine	500

Table 4: List of the hyperparameters selected when fine-tuning BabyLlama-2 on the various SuperGLUE tasks. We do not use early-stopping, since it interfered with BOHB's own early-stopping mechanism. The random seed is 12 for all runs.

Task	Run 1	Run 2
CoLA (MCC)	34.9	31.4
SST-2	85.8	83.5
MRPC (F_1)	82.2	83.8
QQP (F_1)	84.1	84.3
MNLI	74.4	74.3
MNLI-mm	75.3	76.4
QNLI	83.3	83.2
RTE	54.7	61.2
BoolQ	65.9	63.4
MultiRC	64.4	64.9
WSC	57.7	65.4

Table 5: Detailed scores (in %) of the two BabyLlama-2 models on the SuperGLUE tasks. Unless specified otherwise, the listed score is the accuracy. Hyperparameters were optimized for run 1, and then transferred to run 2.

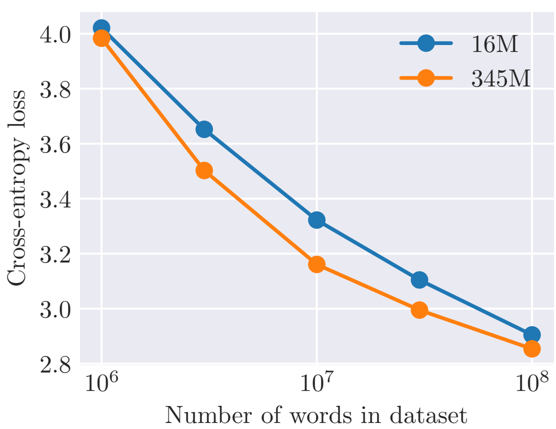


Figure 3: Cross-entropy loss (on the validation split) as a function of dataset size for 16M and 345M models.

16M and 345M models. The 345M model consistently outperforms the 16M model across all dataset sizes, demonstrating that larger models can more efficiently utilize data, hence justifying our choice of the 345M architecture for the final BabyLlama-2 model.

C FineWeb-Edu dataset

Throughout this work, we primarily used the BabyLM-2 dataset. In the early stages, we also experimented with the FineWeb-Edu dataset (Lozhkov et al., 2024), which consists of educational web pages filtered from the FineWeb dataset. We randomly sampled documents containing 20M words (evenly split between the training and validation sets), trained a new tokenizer on this data, and evaluated several variants of the 16M BabyLlama model. The BLiMP scores were consistently lower for models trained on FineWeb-Edu compared to those trained on the BabyLM-2 dataset.[6] We speculate that this lower performance may be due to the limited diversity of examples in FineWeb-Edu, which lacks, for instance, dialogues and non-fiction prose, that are present in BabyLM-2.

References

Loubna Ben Allal, Anton Lozhkov, and Elie Bakouch. 2023. SmolLM - blazingly fast and remarkably powerful. https://huggingface.co/blog/smollm. Accessed: 2023-10-04.

Lukas Biewald. 2020. Experiment tracking with Weights and Biases. Software available from wandb.com.

[6]We do not report specific numbers here since the method for averaging the scores has changed since these experiments were conducted.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts BERT. *arXiv preprint arXiv:2311.02265*.

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. *Computing Research Repository*, arXiv:2404.06214.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Stefan Falkner, Aaron Klein, and Frank Hutter. 2018. BOHB: robust and efficient hyperparameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1436–1445. PMLR.

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.

Jill Gilkerson, Jeffrey A. Richards, Steven F. Warren, Judith K. Montgomery, Charles R. Greenwood, D. Kimbrough Oller, John H. L. Hansen, and Terrance D. Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

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

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 Akyürek, 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. *Preprint*, arXiv:2405.09605.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *Preprint*, arXiv:1907.11692.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. 2024. MobileLLM: Optimizing sub-billion parameter language models for on-device use cases. In *Forty-first International Conference on Machine Learning*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. FineWeb-Edu.

Risto Luukkonen, Ville Komulainen, Jouni Luoma, Anni Eskelinen, Jenna Kanerva, Hanna-Mari Kupari, Filip Ginter, Veronika Laippala, Niklas Muennighoff, Aleksandra Piktus, et al. 2023. FinGPT: Large generative models for a small language. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2710–2726.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models. *arXiv e-prints*, arXiv:2305.16264.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets british national corpus. *arXiv preprint arXiv:2303.09859*.

Inar Timiryasov. 2023. Speed of Llama. `https://timinar.github.io/posts/speed-of-llama/`.

Inar Timiryasov and Jean-Loup Tastet. 2023a. Baby Llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289.

Inar Timiryasov and Jean-Loup Tastet. 2023b. BabyLlama GitHub repository.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *Preprint*, arXiv:1905.00537.

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.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

Chengxu Zhuang, Ethan G Wilcox, Alex Warstadt, and Aaron Mueller. 2024. BabyLM_2024.

Teaching Tiny Minds: Exploring Methods to Enhance Knowledge Distillation for Small Language Models

Hong Meng Yam
Stanford University
hongmeng@stanford.edu

Nathan Paek
Stanford University
nathanjp@stanford.edu

Abstract

In this paper, we build off of the success of the previous BabyLM challenge winner's model, BabyLlama, to explore various methods of enhancing knowledge distillation for small language models. Our main focus is on investigating how small a language model can be while still maintaining competitive performance. We experiment with three main approaches: (1) DistilledGPT-44M, which uses smaller teacher models and a more compact student model compared to BabyLlama; (2) ContrastiveLlama-58M, which incorporates contrastive loss into the knowledge distillation process; and (3) MaskedAdversarialLlama-58M, incorporates adversarial loss into the knowledge distillation process. Using the 10M-word dataset from the BabyLM challenge's strict-small track, we evaluate our models on the BLiMP, EWoK, and GLUE benchmarks. Our results show that effective knowledge distillation can still be achieved with significantly smaller teacher and student models. In particular, our model DistilledGPT-44M is able to achieve better performance than one of last year's winning entries, LTG-BERT, while achieving similar performance but cutting training time by around 70% and parameters by around 25% compared to the other winning entry, BabyLlama.

1 Introduction

Since 2017, transformers have been everywhere in NLP (Vaswani, 2017). Their non-autoregressive nature allows for high parallelization, leading to unprecedented scalability. In recent years, a number of models with trillions of parameters have emerged, such as Google's Switch Transformer (1.6 trillion) and Huawei's PanGu-Σ (1.1 trillion). Models like these or models in the billions demand enormous computational resources and vast swathes of training data. They consume substantial energy, raising concerns about their environmental impact (Bender et al., 2021). The costs of continued scaling are also increasingly prohibitive, highlighting the need for more sample-efficient model architectures.

The 2024 BabyLM challenge (Choshen et al., 2024), by limiting the amount of training data available, in some ways aims to address the computational concerns around large models. However, the scope of the contest focuses more on limiting the amount of training data rather than limiting parameter size or compute; participants have the freedom to use models as large as they want. But last year, the winners of BabyLM (Timiryasov and Tastet, 2023) demonstrated with their model BabyLlama that a small model can outperform models close to an order of magnitude larger on NLP tasks. Model parameter efficiency does not necessarily mean worse results; in fact, in some cases it means better results.

BabyLlama used knowledge distillation and ensemble learning to distill knowledge from two teacher models - GPT2-705M and Llama-360M - to a smaller Llama-58M student model (Hinton et al., 2015). As a model compression technique, knowledge distillation (KD) has several advantages: it only requires access to the teacher model's output logits (not its weights), and it is also model agnostic.

Building on BabyLlama's success, we aim to demonstrate that even smaller teachers and students can achieve competitive performance, further pushing the boundaries of parameter efficiency. We explore the impact of using teachers with fewer parameters and distilling knowledge into even smaller student models. We also explore incorporating different losses into the distillation training, such as contrastive loss and adversarial loss.

Our results suggest that effective knowledge distillation can be achieved with significantly smaller teacher and student models, demonstrating competitive performance even with reduced parameters. We find that our DistilledGPT-44M model, despite

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 302–307
November 15-16, 2024 ©2024 Association for Computational Linguistics

having much fewer parameters, achieves results comparable to the original BabyLlama-58M on key benchmarks. Our experiments with contrastive and adversarial learning techniques in the distillation process, which albeit less promising, reveal interesting trade-offs between different aspects of model performance.

2 Dataset

We use the provided 10M dataset from the strict-small track and build on the BabyLlama repository `https://github.com/timinar/BabyLlama`. Following their preprocessing steps, we apply regex-based cleaning and train a Byte-Pair Encoding tokenizer on the training set. The train and dev sets are split into 128-token chunks, with the model being presented a new random permutation of these chunks in each epoch. Validation loss is computed at the end of each epoch using a fixed, randomly sampled subset of the dev set.

3 Evaluation

Evaluation of model performance was done using the BabyLM evaluation suite (Choshen et al., 2024). This consists of the following benchmarks:

- **BLiMP:** BLiMP (Benchmark of Linguistic Minimal Pairs for English) evaluates language models on their ability to identify grammatical acceptability. It presents pairs of sentences that differ by one linguistic element, testing the model's understanding of 12 areas of English morphology, syntax, and semantics, such as anaphor agreement and filler-gap constructions. It measures how well models assign higher probability to the grammatically correct sentence in each pair.

- **EWoK:** EWoK (Elements of World Knowledge) evaluates language models on their ability to build and apply internal world models. It tests models' understanding of concepts and contexts by presenting them with minimal pairs of scenarios where models must determine the plausibility of context-target combinations. The framework spans 11 knowledge domains.

- **GLUE:** GLUE (General Language Understanding Evaluation) evaluates language models on a variety of natural language understanding tasks. It covers tasks such as sentiment analysis, text similarity, question answering, and textual entailment. LORA finetuning is used for GLUE in this case, though due to computational constraints, this was only evaluated for DistilledGPT-44M, as it was the only one that showed substantial improvement for BLiMP and EWoK. We get the macroaverage by averaging scores across 9 subtasks - all the subtasks in the BabyLM evaluation suite except for CoLA as CoLA only reports matthews correlation scores.

4 Experiments

4.1 Baselines

We first trained GPT2 models in various sizes (18M, 44M, 97M, 705M) and Llama models in various sizes (20M, 60M, 360M) as a baseline and as future teacher models, using same hyperparameters used in the code of BabyLlama (Timiryasov and Tastet, 2023). Model parameters can be found in Table 1.

4.2 DistilledGPT-44M

For our first experiment, we explore the effect of using smaller teacher models and a more compact student model compared to the original BabyLlama configuration.

We use GPT2-44M and Llama-60M as teacher models, both of which are substantially smaller than the GPT2-705M and Llama-360M teachers used in the original BabyLlama. For the student model, we opt for GPT2-44M instead of the Llama-58M used in BabyLlama. This configuration is a significant reduction in the total number of parameters across both teachers and student.

The knowledge distillation process follows the same general approach as BabyLlama. We first train both teacher models (GPT2-44M and Llama-60M) on our dataset. We then train the GPT2-44M student model using a combination of cross-entropy student loss with true labels, and distillation loss between the student's output and each teacher's output. Model architecture for GPT2-44M follows the baseline 44M model.

4.3 ContrastiveLlama-58M

For our second experiment, we bring contrastive loss into the knowledge distillation process. Contrastive learning tries to bring the representations of similar samples closer together while pushing dissimilar samples apart in the embedding space (Chen et al., 2020). For our task, we use contrastive

Hyperparameter	GPT2-18M-all	GPT2-44M-all	GPT2-97M-all	GPT2-705M-all	Llama-20M-all	llama-60M-all	Llama-360M-G10
Hidden dimension size	320	768	768	1536	384	768	1024
Number of layers	2	2	12	24	2	2	24
Number of attention heads	4	8	12	16	4	8	8
Residual dropout	0.0	0.0	0.0	0.1	N/A	N/A	N/A
Attention dropout	0.0	0.0	0.0	0.1	N/A	N/A	N/A
Embedding dropout	0.0	0.0	0.0	0.1	N/A	N/A	N/A
Learning rate	7e-4	7e-4	7e-4	2.5e-4	3e-4	3e-4	3e-4
Batch size	128	128	128	128	128	128	128
Number of epochs	6	6	6	4	4	4	4
Gradient accumulation steps	2	2	2	16	1	1	8
Warmup steps	300	300	300	300	300	300	300
Mixed precision training (fp16)	True	True	True	True	True	True	True

Table 1: Model hyperparameters for baseline models

loss to encourage the student to produce similar hidden representations to the teacher for the same input while distinguishing between representations of different inputs.

We use GPT2-705M and Llama-360M as teacher models and Llama-58M as the student model. The contrastive loss is computed using the N-pair loss formulation, which considers one positive pair and multiple negative pairs in each training iteration. We set N to 32. For each training sample, we generate 31 negative samples by randomly selecting other samples from the same batch. The positive pair consists of the hidden representations of the teacher and student for the same input, while negative pairs are formed by pairing the teacher's representation with the student's representations for different inputs.

We subdivide the overall loss into 39% cross-entropy student loss with true labels, 39% distillation loss, and 22% N-pair contrastive loss computed on the hidden representations of the teacher and student models. This relative weights of loss were chosen through a preliminary linear search for optimal weights by training on a very small subset of data. Model architecture for student model follows that of BabyLlama-58M model.

4.4 MaskedAdversarialLlama-58M

Our next experiment incorporates adversarial learning into the distillation process by implementing the MATE-KD (Masked Adversarial TExt, a Companion to Knowledge Distillation) algorithm (Rashid et al., 2021). MATE-KD enhances traditional knowledge distillation by introducing an adversarial text generator.

The MATE-KD process consists of two main steps:

- Maximization step: A pre-trained masked language model (MLM) generator is trained to perturb the input text by maximizing the divergence between teacher and student logits. This generator learns to create challenging examples that highlight the differences between the teacher and student models.
- Minimization step: The student model is then trained using knowledge distillation on both the original and perturbed training samples, encouraging it to match the teacher's performance on both standard and adversarial inputs.

For our implementation, we use ELECTRA-56M as the generator, pretraining it on our dataset. Our teacher models are GPT2-44M and Llama-60M, both pretrained on our dataset. The student model remains Llama-58M, consistent with our previous experiments. We equally weight cross-entropy student loss with true labels, knowledge distillation loss, and adversarial distillation loss on perturbed samples in our loss function. Model architecture for student model follows that of BabyLlama-58M model.

5 Results

Our results for these 3 experiments can be found in Table 2.

5.1 DistilledGPT-44M

Our DistilledGPT-44M results are encouraging, as they demonstrate that our significantly smaller model configuration can still achieve competitive performance.

From table 2, we can see that DistillGPT-44M manages to outperform both its parent models, GPT2-44M (which scored 58.2 on BLiMP Supplement and 65.6 on BLiMP Filtered) and Llama-60M (which scored 56.7 on BLiMP Supplement and 63.5 on BLiMP Filtered). This shows that

Child Model	Parent Model 1	Parent Model 2	BLiMP Supplement	BLiMP Filtered	EWoK
BabyLlama-58M	GPT2-705M	Llama-360M	59.5	69.8	50.7
ContrastiveLlama-58M	GPT2-705M	Llama-360M	59.3	68.5	50.0
MaskedAdverserialLlama-58M	GPT2-44M	Llama-60M	56.8	65.9	49.6
DistilledGPT-44M	GPT2-44M	Llama-60M	58.8	66.8	50.0

Table 2: Summary of BLiMP filtered, BLiMP supplement and EWOK results for various methods tried for improving knowledge distillation

	BLiMP Supplement	BLiMP Filtered	EWoK
GPT2-18M	55.9	63.7	49.7
GPT2-44M*	58.2	65.6	50.4
GPT2-97M	58.0	66.0	50.6
GPT2-705M	56.7	66.1	50.6
Llama-20M	56.6	62.8	50.2
Llama-60M*	56.7	63.5	49.6
Llama-360M	55.1	68.2	50.5
LTG-BERT	60.8	60.6	48.9
BabyLlama-58M	59.5	69.8	50.7
DistilledGPT-44M	*58.8*	*66.8*	*50.0*

Table 3: Summary of BLiMP Filtered, BLiMP Supplement, and EWoK performance compared to various benchmarks. Our model is in italics, and * represents its teacher models

DistillGPT-44M is able to draw insights from both parents.

This shows that beyond the normal paradigm of a much larger parent model training a student model, we can use collaborative multi-teacher knowledge distillation to create a model that outperforms both parent models.

We also ran finetuned DistilledGPT-44M on GLUE and compared it against BabyLlama-58M baseline results released by BabyLM organizers, and showed that it comparably (Table 3). DistilledGPT-44M excels in tasks requiring nuanced contextual understanding, such as RTE (natural language inference) and WSC (Winograd Schema Challenge), suggesting strong capability in reasoning tasks. While BabyLlama-58M outperforms DistilledGPT-44M on similarity-focused tasks like QQP and sentiment analysis in SST-2, DistilledGPT-44M's competitive scores highlight its efficient handling of complex, context-dependent tasks, even with a smaller parameter set.

BabyLlama-58M demonstrates stronger generalization across a variety of sentence-pair classification tasks, excelling in QNLI, MNLI, and BoolQ. It also outperforms DistilledGPT-44M on CoLA, indicating better linguistic acceptability. However, DistilledGPT-44M's competitive performance in reasoning tasks suggests an efficient and resource-effective model, making it a viable alternative in scenarios where model size is a constraint. These results underscore DistilledGPT-44M's balance of size and performance, standing strong against the larger BabyLlama model in both accuracy and task diversity.

Additionally, the total training time was greatly reduced from the time it took to train BabyLlama-58M. When training on an A5000 GPU, we reduced the total training time from around 10 hours to around 3 hours, which is a more than 3x reduction in training time.

When running a Wilcoxon Ranked-Sum Test on DistilledGPT-44M and BabyLlama-58M for BLiMP, EWoK and GLUE tests seperately, we see that they are **statistically similar** for both BLiMP, EWoK and GLUE, showing that we are able to achieve comparable performance with greatly reduced training times.

Model	MRPC (F1)	RTE	MultiRC	QQP (F1)	QNLI	WSC	MNLI	SST-2	BoolQ	CoLA (MCC)	Macro Avg
DistilledGPT-48M	80.9	55.4	64.9	75.1	77.4	57.7	66.9	75.9	65.3	-0.01	68.8
BabyLlama-58M	82.0	49.6	60.1	83.6	82.8	38.5	72.4	86.2	65.0	2.2	68.9

Table 4: Results of DistilledGPT compared to BabyLlama-58M in GLUE Benchmark

5.2 ContrastiveLlama-58M

Our ContrastiveLlama-58M model show a slight improvement over the baseline GPT2 and Llama models of similar size, and it performs similarly to BabyLlama-58M, with no substantial difference when we perform a Wilcoxon signed rank test. Nonetheless, we currently do not see a benefit to introducing this contrastive loss giving performance remained around the same. We see a trade-off between contrastive learning and traditional knowledge distillation; in future experiments, different weighting schemes for the losses would be interesting to try.

5.3 MaskedAdversarialLlama-58M

Our MaskedAdversarialLlama-58M model shows a decrease in performance compared to both the BabyLlama-58M baseline and our other experiments. The drop is noticeable in the BLiMP Supplement task, where the score is lower than even the baseline GPT2 and the similarly-sized Llama models. This might suggest that the adversarial training might be conflicting with the student model's ability to capture certain linguistic nuances. It could be possible that the generated adversarial examples are too challenging or not representative enough of the task-specific knowledge required for these evaluations. Similarly with our contrastive experiment, trying different weighting schemes for the loss components might help in balancing the trade-off between robustness and task-specific performance in the future.

6 Limitations and Future Work

Although we showed the effectiveness of knowledge distillation with smaller models, we did not thoroughly explore the lower bounds of model size. In future experiments we could investigate even smaller student models or experiment with a wider range of teacher-student size combinations to find the optimal balance between model size and performance.

Additionally, our experiments with contrastive and adversarial learning techniques (ContrastiveLlama-58M and MaskedAdversarialLlama-58M) did not show improvements over the simpler DistilledGPT-44M model. These advanced techniques probably require further refinement or different implementation strategies to be effective: we could try different weighting schemes for loss components in contrastive and adversarial training. Additionally, for the masked adversarial model, the performance of the generator plays a critical role in generating effective perturbed inputs. Using a more powerful MLM generator, rather than the smaller ELECTRA-56M model we used, could improve the adversarial training process and create better perturbations.

7 Conclusion

Herein, we showed that knowledge distillation can be used even with two very simple parents with around the same number of parameters as the child model, to produce a child model which outperforms both parents. We present DistillGPT-44M, which outperforms both the baseline (GPT2) and one of last year's winning entry for the BabyLM challenge LTG-BERT, while maintaining comparable performance to the other winning entry BabyLlama-58M despite reducing number of parameters by around 25% and cutting training time by around 70%.

References

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

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. *Preprint*, arXiv:2404.06214.

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

Ahmad Rashid, Vasileios Lioutas, and Mehdi Rezagholizadeh. 2021. MATE-KD: Masked adversarial TExt, a companion to knowledge distillation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1062–1071, Online. Association for Computational Linguistics.

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. *Preprint*, arXiv:2308.02019.

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

BERTtime Stories: Investigating the Role of Synthetic Story Data in Language Pre-training

Nikitas Theodoropoulos, Giorgos Filandrianos, Vassilis Lyberatos, Maria Lymperaiou and **Giorgos Stamou**

Artificial Intelligence and Learning Systems Laboratory
School of Electrical and Computer Engineering
National Technical University of Athens

nikitastheodorop@gmail.com,
{geofila, vaslyb, marialymp}@ails.ece.ntua.gr,
gstam@cs.ntua.gr

Abstract

We describe our contribution to the `Strict` and `Strict-Small` tracks of the 2nd iteration of the BabyLM Challenge. The shared task is centered around efficient pre-training given data constraints motivated by human development. In response, we study the effect of synthetic story data in language pre-training using *TinyStories*: a recently introduced dataset of short stories. Initially, we train GPT-Neo models on subsets of *TinyStories*, while varying the amount of available data. We find that, even with access to less than 100M words, the models are able to generate high-quality, original completions to a given story, and acquire substantial linguistic knowledge. To measure the effect of synthetic story data, we train LTG-BERT encoder models on a combined dataset of: a subset of *TinyStories*, story completions generated by GPT-Neo, and a subset of the *BabyLM* dataset. Our experimentation reveals that synthetic data can occasionally offer modest gains, but overall have a negative influence on linguistic understanding. Our work offers an initial study on synthesizing story data in low resource settings and underscores their potential for augmentation in data-constrained language modeling. We publicly release our models and implementation on our GitHub [1].

1 Introduction

As the performance of modern Language Models (LMs) increases, enabling remarkable feats of language understanding and reasoning, so do their demands in computational resources and training data (Hoffmann et al., 2022). For example, the recently released Llama 3 (Dubey et al., 2024) has 405B parameters and was pre-trained on 15.6T tokens, on 6K H100 GPUs. In contrast, children are only exposed to no more than 100 million words by age 13 (Gilkerson et al., 2017), demonstrating exceptional learning efficiency compared to state-of-the-art LMs. This need for ever-increasing data and compute casts doubts on the cognitive plausibility of the current LM training regimes, and raises ecological and ethical concerns, such as democratic access to research for industry and research groups with modest resources.

To address these issues, the BabyLM challenge (Warstadt et al., 2023a; Choshen et al., 2024) invites participants to work on cognitive modeling and efficient LM pre-training, given data limitations inspired by human development. This year's iteration of the challenge features three experimental tracks: a `Strict` track with a budget of 100M words, a `Strict-Small` track with a budget of 10M words, and a `Multimodal` track with a word budget of 100M words and unlimited visual input. A major change compared to last year's challenge is allowing participants to construct their own training data. In the following sections, we present our contributions to the `Strict` and `Strict-Small` tracks.

Our research draws inspiration from recent advancements in Small Language Models (SLMs) for text generation, as explored in *TinyStories* (Eldan and Li, 2023). In this influential work, the authors demonstrate that training on a synthetic dataset of simple stories can enable SLMs to produce creative, high-quality generations, which are novel with respect to the original training dataset. We hypothesize that for the small data regimes of the BabyLM challenge, *augmenting* the initial training corpus with synthetic data of high quality can provide models with unseen linguistic contexts, and as a result improve language understanding. To test our hypothesis, we first extend previous work by

[1] https://github.com/nikitas-theo/BERTtimeStories

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 308–323
November 15-16, 2024 ©2024 Association for Computational Linguistics

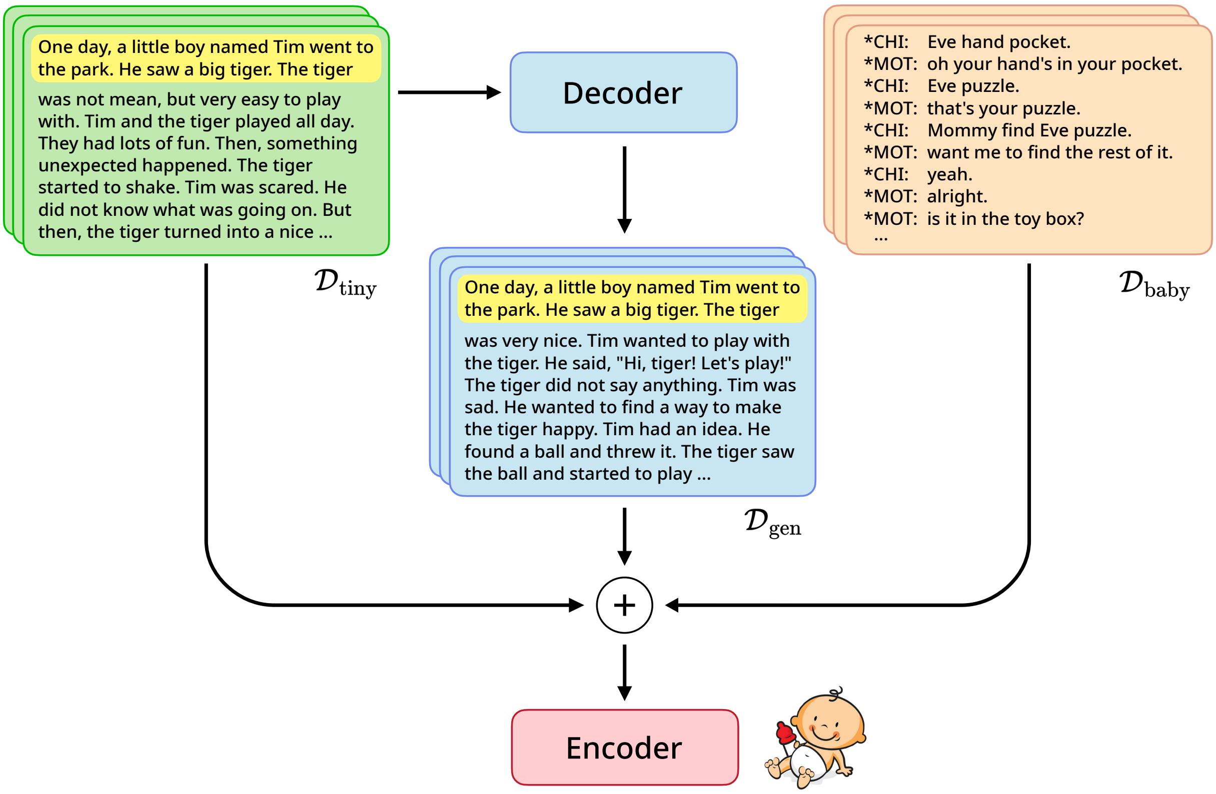


Figure 1: Illustration of our proposed methodology for *BERTtime Stories*. We use a subset of the *TinyStories* dataset ($\mathcal{D}_{\text{tiny}}$) (Eldan and Li, 2023), to train a decoder transformer for data augmentation. We prompt the decoder with the short stories from $\mathcal{D}_{\text{tiny}}$ and create a dataset of model generations ($\mathcal{D}_{\text{gen}}$): each story (green) is truncated and used as a prompt (yellow), with the model generating an alternate completion (blue). We supplement the two datasets with a subset of the *BabyLM* dataset ($\mathcal{D}_{\text{baby}}$), released by Choshen et al. (2024), and train an encoder model on the combined data. Finally, we evaluate the linguistic proficiency of the encoder using the challenge benchmarks.

Eldan and Li (2023), investigating generative performance with limited training data. We then train encoder transformer models on a diverse dataset, and measure the effect of synthetic data on linguistic proficiency.

In technical terms, following Eldan and Li (2023), we propose to train a GPT-Neo decoder (Black et al., 2021) on *TinyStories*, and then use it to *generate data* for the training of a final encoder model. This poses some initial challenges, as the size of the original *TinyStories* dataset exceeds the challenge limits, with around 373M words. As a result, we experiment with training GPT-Neo models while varying the *amount of available data* and evaluating their generative performance, keeping the model architecture fixed in the process. Our investigation of generative performance is complementary to the original work, which measures the effects of varying the depth and size of the model architectures. Our findings demonstrate that even in low data regimes of $\leq$ 100M words, GPT-Neo models can acquire impressive grammatical understanding, and generate stories of *high quality* and *diversity*, comparable to models trained on the full dataset. For our evaluation see Section 4.1.

Next, we investigate the effect of the synthetic story data on language pre-training. Figure 1 illustrates our methodology. We select a small subset of *TinyStories*, train a GPT-Neo model, and use it to generate alternate completions to the stories in the training set. We then create a combined dataset consisting of: 1 the subset of *TinyStories* used for GPT-Neo training, 2 the generated data, 3 a sample of the *BabyLM* dataset (Choshen et al., 2024). With the combined dataset, we train an LTG-BERT (Samuel et al., 2023) model, choosing this architecture as it demonstrated superior performance in the text-only tracks of last year. We compare the performance of our models against a variety of baselines, trained with and without the use of synthetic data. Our results show that a simple application of synthetic story data for language pre-training results in *marginal* or even *negative* gains. Nevertheless, given the high generative performance of the GPT-Neo models, we believe that more research is needed to fully explore and exploit their potential.

Contribution We list our contributions below:

- We investigate the generative and linguistic abilities of GPT-Neo models trained on *TinyStories* while varying the amount of available data. We show that even with limited data, these models can produce generations of high quality, offering new insights into the capabilities of SLMs in low data regimes.

- We investigate the effect of generated data on the pre-training of encoder LMs in a constrained data setting. We conduct an extensive evaluation with different training schemes and baselines. Our experiments demonstrate the potential of data augmentation to enhance the linguistic capabilities of low resource LMs.

2 Related work

Previous BabyLM Iteration Data Augmentation techniques were shown to be beneficial in the previous year's challenge (Warstadt et al., 2023b). Specifically, ChapGPT (Jumelet et al., 2023) uses regex patterns to extract common phrases from GLUE tasks, and then harnesses these patterns to generate follow-up questions that serve as additional training data. In the Contextualizer paper (Xiao et al., 2023), extra training samples are created by dynamically combining chunks of texts from different contexts during training. Another approach named Baby's CoThought (Zhang et al., 2023) utilizes a Large Language Model (LLM) to reformat unrelated sentences from the corpus into coherent paragraphs, thereby improving performance, albeit in defiance of data constraints.

Language Models for Data Augmentation In recent years, LLMs have been increasingly leveraged for data augmentation in various domains (Ding et al., 2024). Notably, Dai et al. (2023) introduced ChatGPT as a tool for generating realistic text samples from a combination of real and artificial data, enhancing training datasets. Similarly, transformer architectures, including decoder (GPT-2, Radford et al., 2019), encoder (BERT, Devlin et al., 2019), and seq2seq (BART, Lewis et al., 2020) models have been explored for augmentation (Kumar et al., 2020). In the work of Yoo et al. (2021), GPT-3 (Brown et al., 2020) was used to mix real and synthetic text samples for robust data augmentation. Moreover, decoder models have been successfully employed to generate training data for encoders, yielding significant improvements in zero-shot learning (Meng et al., 2022).

Small Language Models The recent study by Eldan and Li (2023) highlighted that Small Language Models (SLMs), can outperform larger ones by leveraging high-quality synthetic training data, demonstrating fluency, coherence, and creativity despite having fewer parameters. This trend is further supported by work in sequential recommendation, where small models are effectively employed for task-specific purposes (Xu et al., 2024). Additionally, Bergner et al. (2024) utilize a pre-trained LLM to encode prompt tokens, using these representations to guide a smaller LM for more efficient response generation.

3 Methods

We describe our data augmentation method using synthetic story data, as illustrated in Figure 1.

3.1 Datasets

Our work is built on two datasets: ① *TinyStories* – denoted as $\mathcal{D}_{\text{tiny}}$, a collection of synthetic short stories with simple language, ② the *BabyLM* dataset – denoted as $\mathcal{D}_{\text{baby}}$, created to be a developmentally plausible pre-training corpus. For any dataset $\mathcal{D}_{\text{data}}$, we also denote a version of the data with m million words as $\mathcal{D}_{\text{data-m}}$. We describe the datasets below:

***BabyLM* dataset** The *BabyLM* dataset ($\mathcal{D}_{\text{baby}}$), released by Warstadt et al. (2023a); Choshen et al. (2024), consists of a diverse set of texts and is constructed with the goal of simulating the linguistic input that a child receives throughout its development. It contains a high proportion of spoken language and includes, among others, excerpts from children's books, dialogue, child-directed speech, and Wikipedia articles. Both 100M and 10M versions of the dataset were released, for the `Strict` and `Strict-Small` tracks respectively. Details about the dataset structure are provided in Appendix A.

***TinyStories* dataset** Introduced by Eldan and Li (2023), *TinyStories* ($\mathcal{D}_{\text{tiny}}$) is a synthetic dataset, featuring a collection of short stories constructed by prompting GPT-3.5 and GPT-4 (OpenAI et al., 2024). The dataset was created to preserve all the core elements of natural language, such as grammar and reasoning, while exhibiting limited diversity and size. More specifically, the stories are 2-3 paragraphs long and follow simple plots and themes. In addition, the dataset contains a restricted vocabulary and in general is intended to be on the level of understanding of 3-4 year old children. The initial version of the dataset (V1), generated by both GPT-3.5 and GPT-4, contains approximately 373M words. A second version (V2) was later released, with stories generated only by GPT-4 and around 440M words. We use this version in all our experiments.

3.2 Data Generation

We describe the creation of the synthetic story dataset $\mathcal{D}_{\text{gen}}$. To generate the data, we first train a decoder model (GPT-Neo) on a subset of *TinyStories* denoted as $\mathcal{D}_{\text{tiny-m}}$. We truncate the stories in $\mathcal{D}_{\text{tiny-m}}$ to construct prompts and generate alternate completions using our model.

We start by restricting the size m of the subset, taking into account two factors: the need for adequate *diversity* in the final corpus, and the need to ensure *high-quality* generations. Given the assumption that generation quality scales with dataset size, we want to select a big enough size m for $\mathcal{D}_{\text{tiny-m}}$ to enable high-quality generations from our trained models. At the same time, we want to leave the necessary room in our word budget for including a sufficiently large portion of the *BabyLM* dataset in the final training. This will ensure that our models are exposed to both a large vocabulary and a variety of word contexts. Intuitively, we aim to ensure that our pre-training data is diverse, as children learn from multiple sources of input.

To address this trade-off, we sample from *TinyStories*, creating a collection of subsets of varying sizes, $\mathcal{D}_{\text{tiny-m}} : m \in \{5, 10, 25, 50, 75, 100\}$M (millions of words). For each subset, we train a GPT-Neo model and evaluate its generative and linguistic abilities. In our evaluation, we leverage metrics for grammatical understanding, diversity, and generation quality; our metrics are introduced in Section 3.4. For each of the `Strict` and `Strict-Small` tracks, we select a subset $\mathcal{D}_{\text{tiny-m}}$ and a corresponding GPT-Neo model trained on it, based on our evaluation metrics and the above criteria. To construct $\mathcal{D}_{\text{gen}}$, for each story in $\mathcal{D}_{\text{tiny-m}}$, we truncate the story to 15%-30% of its size and use it to *prompt* the model for generation. We opt for using a smaller proportion of the original story to avoid duplication, given that stories in $\mathcal{D}_{\text{tiny-m}}$ will already be in the combined corpus for the training of the encoder transformer.

Regarding the generation process, we experiment with two methods: greedy decoding and nucleus sampling (Holtzman et al., 2020). During sampling, we generate k completions from our models for each prompt. To limit repetition between the k generations (and avoid wasting FLOPs), we calculate Self-BLEU (Section 3.4) for a set of values of k, and select the ones that best balance diversity and the total amount of additional training data.

3.3 Final Corpus Creation

For each of the `Strict` and `Strict-Small` tracks, we have created $\mathcal{D}_{\text{tiny-m}}$, and $\mathcal{D}_{\text{gen}}$ as previously described. We now create the combined dataset $\mathcal{D}_{\text{comb}}$, used to train the encoder transformer. We allocate our remaining word budget to a subset of the *BabyLM* dataset ($\mathcal{D}_{\text{baby-b}}$), created by sampling randomly from *BabyLM* on the document level. We leave sampling methods that account for the content of the documents for future work. For the `Strict` / `Strict-Small` tracks, the size b of $\mathcal{D}_{\text{baby-b}}$ is chosen such that: $b + m \leq 100$M / 10M. We now construct $\mathcal{D}_{\text{comb}}$ by combining all the datasets $\mathcal{D}_{\text{comb}} = (\mathcal{D}_{\text{tiny-m}}, \mathcal{D}_{\text{baby-b}}, \mathcal{D}_{\text{gen}})$. We employ a masked language modeling objective to train an encoder transformer on $\mathcal{D}_{\text{comb}}$, with the LTG-BERT architecture (Samuel et al., 2023).

3.4 Evaluation

For evaluating the encoder transformers we use the evaluation suite of the challenge, consisting of three evaluation benchmarks: BLiMP, (Super)GLUE, and EWoK, each broadly evaluating language proficiency, general language understanding, and world knowledge. We note that the challenge benchmarks constitute filtered versions (Warstadt et al., 2023b), rendering our results incomparable with full data evaluations. For the decoder models, we use EWoK and BLiMP, and also introduce some additional evaluation procedures: specifically, Self-BLEU evaluates diversity, and an LLM-assisted evaluation measures generation quality. We explain each of the evaluation benchmarks below.

BLiMP The Benchmark of Linguistic Minimal Pairs (BLiMP), introduced by Warstadt et al. (2019), is a set of tasks designed to evaluate the linguistic knowledge of LMs. It consists of pairs of minimally different sentences covering various grammatical phenomena in syntax, morphology, and semantics. The model under evaluation has to assign a higher probability to the correct sentence in each pair. We also evaluate on BLiMP Supplement (Supp.), released by Warstadt et al. (2023a), which includes additional grammatical phenomena. For both BLiMP and BLiMP Supplement, we measure performance by calculating the average accuracy across all of their evaluation tasks.

(Super)GLUE The General Language Understanding Evaluation (GLUE) benchmark (Wang, 2018), assesses model performance across a wide range of natural language understanding (NLU) tasks. SuperGLUE (Wang et al., 2019), was later introduced to offer a more challenging set of tasks. We employ a total of 10 text classification tasks from both benchmarks, which include: question answering (BoolQ, MultiRC), sentiment classification (SST-2), paraphrase detection (MRPC, QQP), linguistic acceptability (CoLA), commonsense reasoning (WSC), and natural language inference (MNLI, QNLI, RTE). Performance on (Super)GLUE is calculated by averaging accuracies across all tasks except for QQP and MRPC, where we use the F1-score, and CoLA, where we use the Matthews Correlation Coefficient – MCC.

EWoK Elements of World Knowledge (EWoK) (Ivanova et al., 2024) assesses an LM's ability to understand and model world knowledge. It evaluates how well a model can connect a target text to either an appropriate or mismatched context, emphasizing key concepts such as social dynamics and spatial relationships. Both the contexts and targets are framed as minimally contrasting pairs, with customizable elements like objects, agents, and locations. During evaluation, the model needs to assign a higher probability to the correct context and target text pair. We report average accuracy across all the benchmark's tasks.

Self-BLEU To measure the diversity of generated stories, we utilize the Self-BLEU score (Zhu et al., 2018). Given a generated collection, we calculate the BLEU score with one generation as the hypothesis and the others as reference, evaluating how similar it is to the rest. We define Self-BLEU as the average of all the BLEU scores in the corpus. The metric is defined on a continuous scale within $[0, 1]$, where higher scores indicate less diversity.

LLM Evaluation To provide a comprehensive evaluation of our decoder models' generative abilities, we follow the approach of Eldan and Li (2023) and employ a LLM, prompting it with the story completions, and asking it to assess them in terms of *Grammar*, *Creativity*, and *Consistency* with the story's beginning, on a scale from 1 to 10. The original evaluation by Eldan and Li (2023) used GPT-4, we instead leverage Claude-3.5 Sonnet (Anthropic, 2024)[2], which better aligned with our available resources. Evaluation details are presented in Section 4.1, while the prompt is included in Appendix E.

4 Experiments

Experimental Setup We conduct our experiments on a shared GPU cluster of 8 Nvidia V100 16 GB GPUs, and additionally evaluate our models on an Nvidia RTX-3090 24 GB GPU. All our models are trained using the PyTorch (Paszke et al., 2019) and HuggingFace (Wolf et al., 2019) libraries. For our evaluations of BLiMP, EWoK, and (Super)GLUE we build upon the official evaluation pipeline released by the challenge organizers (Gao et al., 2023; Choshen et al., 2024).

4.1 TinyStories & GPT-Neo Evaluation

Regarding the decoder used for the generation, we select one of the best-performing GPT-Neo architectures from Eldan and Li (2023) [3]. All our trained GPT-Neo models share the same hyperparameters, except for weight decay, dropout, and vocabulary size, which are tuned to the specific data size. We built upon a similar training scheme as the authors, with added regularization for our low data regime. Hyperparameters and details about the architecture are included in Appendix C. We opt to train on the latest version of the *TinyStories* data (V2), generated by prompting GPT-4; the full unsampled dataset contains $\sim$ 440M words. Throughout our evaluation, we also report results for the original model released by the authors, trained on the first version of the dataset (V1) with $\sim$ 373M words.

In the following paragraphs, we conduct a thorough analysis of the relationship between the linguistic competency of GPT-Neo models trained on subsets of *TinyStories*, and the size of their training dataset $|\mathcal{D}_{\text{tiny-m}}|$. We experiment with various sizes for the *TinyStories* subsets $\mathcal{D}_{\text{tiny-m}}$: $m \in \{5, 10, 25, 50, 75, 100\}$M (millions of words). From our experiments we draw insights about the abilities of generative LMs on low data regimes. This evaluation will also motivate our selection of

[2] Model version: `claude-3-5-sonnet-20240620`.
[3] https://huggingface.co/roneneldan/TinyStories-33M

the *TinyStories* subset $\mathcal{D}_{\text{tiny}}$ used for generating the dataset $\mathcal{D}_{\text{gen}}$ and for training the final encoder.

As an initial proxy of the language competency of the GPT-Neo decoders, we measure performance on BLiMP, its supplement (Supp.), and EWoK. Results are presented in Table 1. We notice that 50M words appear to be a cutoff point, with notable drops in performance for data sizes less than that. Based on this, we select $\mathcal{D}_{\text{tiny-50M}}$ for the `Strict` track, and $\mathcal{D}_{\text{tiny-5M}}$ for the `Strict-Small` track. Importantly, we do not include the LLM evaluation (presented below) in this decision process, as it would invalidate our imposed data constraints. We leave further experimentation on the subset data sizes for the `Strict-Small` track for future work. A second observation concerns the 100M words model, which achieves the top score on BLiMP, shared by the 373M model by Eldan and Li (2023). This result agrees with the findings of Zhang et al. (2021), demonstrating that 100M words are enough to attain substantial grammatical knowledge.

Train Data	BLiMP ↑	Supp. ↑	EWoK ↑
5M	55.5	53.8	51.1
10M	58.4	51.6	51.9
25M	59.9	55.1	52.4
50M	62.8	52.8	53.0
75M	64.0	54.8	53.4
100M	**64.8**	50.8	53.1
440M (V2)	64.6	55.0	53.9
373M (V1) [4]	**64.8**	**60.9**	**54.0**

Table 1: Evaluation results for GPT-Neo models trained on *TinyStories* with various amounts of data. We report *accuracy* for all benchmarks. As the amount of data decreases, the BLiMP and EWoK scores generally decrease as well. In contrast, the BLiMP supplement score demonstrates more variance.

The aforementioned scores give us evidence about the grammatical understanding (BLiMP) and world knowledge (EWoK) of our models, but leave out two important areas of generative performance, mainly: ① the *diversity* and ② the *quality* of generations. We focus on these two metrics in the following paragraphs. Apart from the quantitative scores, in Appendix B we also include the generations of all the GPT-Neo models for the *TinyStories* example illustrated in Figure 1.

Evaluating Generation Quality Evaluating the quality of generations for open-ended generation tasks is challenging, as most common evaluation paradigms expect structured output, and measure fidelity towards a set of reference texts. To address this, we adopt the evaluation method proposed by Eldan and Li (2023), and prompt an LLM to evaluate the stories generated by our models. In our experiments, we use Claude-3.5 Sonnet.

We harness a set of 44 manually constructed prompts [5] containing the beginning of a story, and generate 10 completions for each of our models, sampling with a temperature of 1. We then provide the LLM with the beginning of the story and the model's completion, and ask it in turn to evaluate the model's response along three axes: (a) *Grammar*, (b) *Creativity*, and (c) *Consistency* with the beginning of the story. Additionally, we ask it to classify the story in different age groups, ranging from 3 (or under) to 16 years old. Scores are given on a scale of 1 to 10, and are averaged across stories and completions. The final results are presented in Table 2: we notice that limiting the training data, up to even 25M words, results in a minor decrease of performance across all three metrics. This indicates that the quality of the model generations is retained in the small data regime. Additionally, the 100M words decoder achieves impressive scores in all categories, and outperforms all other models in the *Consistency* metric – demonstrating that 100M words is enough for robust generative performance.

Evaluating Generation Diversity To measure diversity, we utilize Self-BLEU (Zhu et al., 2018), which has been used before as a measure of the diversity of generated data (Holtzman et al., 2020). For each model, we sample 100 stories from the training set and truncate them to 15%-30%, prompting the model to generate an alternate completion to the story's beginning. When sampling from the model, a greedy decoding strategy is employed. We report Self-BLEU scores, scaled to $[0, 100]$, for the set of 100 completions in Table 2 (higher scores correspond to less diverse generations). Our results indicate that models with limited training data can achieve high diversity, while at the same time maintaining generation quality, as demonstrated by the scores of models trained on 25M and 50M words.

4.2 Data Generation

We now describe the creation of the combined dataset $\mathcal{D}_{\text{comb}} = (\mathcal{D}_{\text{tiny-m}}, \mathcal{D}_{\text{baby-b}}, \mathcal{D}_{\text{gen}})$, leveraged for training an encoder LM. For

[4] Model released by Eldan and Li (2023).

[5] https://huggingface.co/datasets/roneneldan/TinyStories

Train Data	Gr. ↑	Cr. ↑	Cons. ↑	SB ↓
5M	4.56	4.99	3.37	38.6
10M	5.31	5.34	3.98	38.3
25M	6.00	5.65	4.55	34.6
50M	6.01	5.53	4.54	33.0
75M	6.08	5.50	4.49	37.1
100M	6.17	5.57	**4.78**	39.8
440M (V2)	5.88	5.53	4.49	37.3
373M (V1)	**6.24**	**5.73**	4.70	**29.6**

Table 2: Results on the evaluation of our models by Claude-3.5 Sonnet. We instruct the LLM to access generative performance along three categories: Grammar (Gr.), Creativity (Cr.), Consistency (Cons.). We also include Self-BLEU (SB), measuring generation diversity.

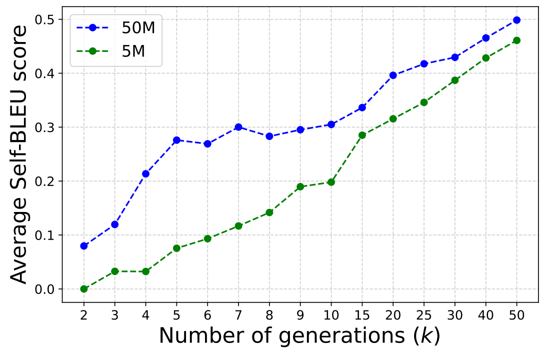


Figure 2: We generate 50 completions for 100 prompts with the GPT-Neo models trained on $\mathcal{D}_{\text{tiny-5M}}$, $\mathcal{D}_{\text{tiny-50M}}$. We plot the average self-BLEU score across prompts, as the number of generations per prompt (k) increases.

brevity, details are given below only for the `Strict-Small` track; the same process is followed for the `Strict` track. As discussed in Section 4.1, we choose a subset of 5M words from *Tinystories* ($\mathcal{D}_{\text{tiny-5M}}$), and use it to train a GPT-Neo model. This model is then employed to generate the dataset $\mathcal{D}_{\text{gen}}$. We adapt the beginning of each story (15%-30%) in the training set $\mathcal{D}_{\text{tiny-5M}}$ as a prompt, and task the decoder to generate alternative completions. We experiment with different generation techniques, including greedy generation – $\mathcal{D}_{\text{gen-greedy}}$, and nucleus sampling – $\mathcal{D}_{\text{gen-nucleus-k}}$, where k is the number of generations per prompt. Finally, the two datasets are combined with a subset of the *BabyLM* dataset ($\mathcal{D}_{\text{baby-5M}}$), ensuring a total size within the 10M word limit, to form $\mathcal{D}_{\text{comb}}^{\text{10M}} = (\mathcal{D}_{\text{tiny-5M}}, \mathcal{D}_{\text{baby-5M}}, \mathcal{D}_{\text{gen}})$.

In order to select k for nucleus sampling, we leverage the Self-BLEU score. We sample 100 stories from $\mathcal{D}_{\text{tiny-5M}}$ and use their beginning (15%-30%) to generate 50 completions for each prompt with $p = 0.95$. For each value of $k \in \{2, 3, ..., 50\}$ we calculate Self-BLEU among the group of generations S_k. Our goal is to examine how diverse the different generations are for the same prompt, as the number of generations (k) increases. Figure 2 depicts the *average* Self-BLEU across all prompts. Based on the presented results, we choose to experiment with $k = 5$ and $k = 10$, as a satisfactory balance between diversity and added dataset size.

4.3 Training LTG-BERT

Following the creation of the combined corpus $\mathcal{D}_{\text{comb}}$, we employ it to train an LTG-BERT (Samuel et al., 2023) encoder module. Our training procedure is based on the source code released by the authors[6], prompting our selection of similar hyperparameters (Appendix C), adapted for our specific infrastructure and available compute. Moreover, our experiments are conducted with minimal hyperparameter optimization. In order to assess the effect of data augmentation on final performance, we train a variety of baselines, ablating over the pre-training dataset of our models and keeping all other training conditions constant. Specifically, for a given track, all the models share the same hyperparameters and amount of FLOPs, ensuring a fair comparison. Our baselines are described below.

Baselines For the `Strict-Small` track, we establish baselines by training LTG-BERT models using 10M words from the *BabyLM* – $\mathcal{D}_{\text{baby-10M}}$ and *Tinystories* – $\mathcal{D}_{\text{tiny-10M}}$ datasets respectively. Additionally, we train an encoder using a combination of 5M words from each one of the two datasets – $\mathcal{D}_{\text{baby-5M}} + \mathcal{D}_{\text{tiny-5M}}$. These models serve as benchmarks against which we assess the performance of models trained with various configurations of generated data, aiming to evaluate the effectiveness of data augmentation. The same methodology is applied consistently to the `Strict` track as well. Here, we train encoders with 100M words from each dataset separately, as well as in a combined setting, utilizing 50M words from each dataset. We also include results for the challenge baselines – LTG-BERT (Samuel et al., 2023) and BabyLlama (Timiryasov and Tastet, 2023). We emphasize that these models are trained with different hyperparameters than those in our controlled setting. Notably, the LTG-BERT model released by the organizers was trained for ~ 20 epochs on the `Strict` track,

[6] https://github.com/ltgoslo/ltg-bert

Model	Training Data	Total	BLiMP	Supp.	EWoK	GLUE	Avg.
LTG-BERT	$\mathcal{D}_{\text{baby-10M}}$	10M	60.6	60.8	47.6	60.3	57.3
BabyLlama	$\mathcal{D}_{\text{baby-10M}}$	10M	69.8	59.5	50.7	63.3	60.8
LTG-BERT (**ours**)	$\mathcal{D}_{\text{baby-10M}}$	10M	62.8	**63.7**	51.2	71.0	**62.2**
	$\mathcal{D}_{\text{tiny-10M}}$	10M	59.8	54.2	**52.2**	67.0	58.3
	$\mathcal{D}_{\text{tiny-10M}} + \mathcal{D}_{\text{gen-greedy}}$	20M	58.7	57.8	48.9	67.1	58.1
	$\mathcal{D}_{\text{baby-5M}} + \mathcal{D}_{\text{tiny-5M}}$	10M	62.6	60.7	51.5	**71.2**	61.5
	$\mathcal{D}_{\text{baby-5M}} + \mathcal{D}_{\text{tiny-5M}} + \mathcal{D}_{\text{gen-greedy}}$	15M	62.1	60.2	50.4	70.6	60.8
	$\mathcal{D}_{\text{baby-5M}} + \mathcal{D}_{\text{tiny-5M}} + \mathcal{D}_{\text{gen-nucleus-1}}$	15M	62.5	62.3	48.8	69.5	60.8
	$\mathcal{D}_{\text{baby-5M}} + \mathcal{D}_{\text{tiny-5M}} + \mathcal{D}_{\text{gen-nucleus-1}}$ † ⋆	15M	**63.2**	59.3	50.4	71.1	61.0
	$\mathcal{D}_{\text{baby-5M}} + \mathcal{D}_{\text{tiny-5M}} + \mathcal{D}_{\text{gen-nucleus-5}}$	33M	62.4	60.1	50.7	69.4	60.6
	$\mathcal{D}_{\text{baby-5M}} + \mathcal{D}_{\text{tiny-5M}} + \mathcal{D}_{\text{gen-nucleus-10}}$	56M	61.0	58.4	50.1	69.5	59.8

Table 3: Model performance for the 10M word `Strict-Small` track.

compared to our setting of $\sim$ 27 epochs (20K steps for both tracks).

Balanced Training While increasing the number of generated texts in the LTG-BERT training set ($\mathcal{D}_{\text{comb}}$), we also modify the distribution of *TinyStories* and *BabyLM* samples that the model encounters during training. This could affect the model's performance, as it becomes more finely tuned to *TinyStories*. To counter this effect, we experiment with a training variation where we balance the number of samples from both datasets. Specifically, samples in each batch are drawn with equal probability from both *TinyStories* – which includes both original and generated texts – and *BabyLM*. This method ensures that the model is exposed to an equal number of samples from each dataset throughout training. The dagger symbol † in the results denotes use of this strategy.

5 Results

We present the final evaluation results for the `Strict-Small` and `Strict` tracks at Table 3 and Table 4, respectively. The ⋆ symbol denotes the submitted model for this track.

Strict-Small Track In the `Strict-Small` track, comparing the results of $\mathcal{D}_{\text{baby-10M}}$ with $\mathcal{D}_{\text{tiny-10M}}$ reveals, as expected, that the *BabyLM* dataset is more beneficial for language pre-training compared to *TinyStories*. The performance metrics for *TinyStories* are consistently lower, except in the case of EWoK. Interestingly, replacing half of the *BabyLM* dataset with data from *TinyStories* only slightly affects the model's performance. However, as we add more instances of the synthetic story data, the positive impact of the *BabyLM* dataset begins to wane, leading performance to approach that of $\mathcal{D}_{\text{tiny-10M}}$ where *BabyLM* was not used at all. This suggests that training is over-influenced by the increased amount of *TinyStories* data. To mitigate this effect, we experimented with equally distributing the samples from the two datasets in a batch. This approach positively impacts the model's performance. Notably for BLiMP, this setup slightly surpasses the performance of the model trained solely on $\mathcal{D}_{\text{baby-10M}}$, resulting in the best score overall. Further, when compared to other data augmentation scenarios, the performance on GLUE is increased.

Moreover, an interesting observation concerns the sampling technique used for augmenting the data. Changing the sampling strategy from greedy decoding to nucleus sampling positively affects the model's performance on the BLiMP and BLiMP Supp. benchmarks, while negatively impacting performance on EWoK and GLUE. This discrepancy is likely due to the nature of the datasets themselves. BLiMP focuses on evaluating grammatical understanding, while the increased diversity from nucleus sampling exposes the model to a wider range of linguistic structures and syntactic variations, thereby improving performance. Conversely, EWoK and GLUE require semantic coherence and factual consistency, where the increased diversity from nucleus sampling may introduce noise and less coherent narratives, potentially confusing the model, and degrading performance. Therefore, while more diverse stories benefit syntactic evaluation tasks such as those in BLiMP, they may not be as useful for semantic or knowledge-based tasks such as those included in EWoK and GLUE.

Strict Track Interestingly, for the `Strict` track we notice that data augmentation has a positive effect on the BLiMP and EWoK benchmarks. Specifically, adding the $\mathcal{D}_{\text{gen-greedy}}$ dataset, results in increased performance compared to the base-

Model	Training Data	Total	BLiMP	Supp.	Ewok	GLUE	Avg
LTG-BERT	$\mathcal{D}_{\text{baby-100M}}$	100M	69.2	66.5	50.2	68.4	63.6
BabyLlama	$\mathcal{D}_{\text{baby-100M}}$	100M	73.1	60.6	52.1	69.0	63.7
LTG-BERT (**ours**)	$\mathcal{D}_{\text{baby-100M}}$	100M	64.0	**67.6**	47.3	**74.0**	**63.2**
	$\mathcal{D}_{\text{tiny-100M}}$	100M	61.2	63.2	48.0	70.6	60.8
	$\mathcal{D}_{\text{tiny-100M}} + \mathcal{D}_{\text{gen-greedy}}$	200M	61.1	59.6	48.7	69.1	59.6
	$\mathcal{D}_{\text{tiny-50M}} + \mathcal{D}_{\text{baby-50M}}$	100M	65.5	65.6	47.2	71.0	62.3
	$\mathcal{D}_{\text{tiny-50M}} + \mathcal{D}_{\text{baby-50M}} + \mathcal{D}_{\text{gen-greedy}}$	150M	**66.6**	63.3	**49.7**	71.8	62.8
	$\mathcal{D}_{\text{tiny-50M}} + \mathcal{D}_{\text{baby-50M}} + \mathcal{D}_{\text{gen-nucleus-1}}\star$	150M	65.6	65.0	49.3	72.7	63.1
	$\mathcal{D}_{\text{tiny-50M}} + \mathcal{D}_{\text{baby-50M}} + \mathcal{D}_{\text{gen-nucleus-1}}\dagger$	150M	65.2	63.5	49.0	72.6	62.6
	$\mathcal{D}_{\text{tiny-50M}} + \mathcal{D}_{\text{baby-50M}} + \mathcal{D}_{\text{gen-nucleus-5}}$	350M	65.4	64.4	45.9	69.8	61.4
	$\mathcal{D}_{\text{tiny-50M}} + \mathcal{D}_{\text{baby-50M}} + \mathcal{D}_{\text{gen-nucleus-10}}$	600M	63.7	63.3	49.2	69.5	61.4

Table 4: Model performance for the 100M word `Strict` track.

lines trained on $\mathcal{D}_{\text{tiny-100M}}$ and $\mathcal{D}_{\text{baby-100M}}$, as well as a mixture of the two ($\mathcal{D}_{\text{tiny-50M}} + \mathcal{D}_{\text{baby-50M}}$). Additionally, the $\mathcal{D}_{\text{tiny-50M}} + \mathcal{D}_{\text{baby-50M}}$ combination is outperformed by both the $\mathcal{D}_{\text{gen-greedy}}$ and $\mathcal{D}_{\text{gen-nucleus-1}}$ models, suggesting that synthetic data can offer modest gains in the `Strict` scenario.

As with the `Strict-Small` track, increasing the size of the *TinyStories* dataset negatively affects the performance of the models, approaching that of the model trained solely on $\mathcal{D}_{\text{tiny-100M}}$. However, in this case, balancing the datasets does not improve the model's performance. In the larger 100M word dataset, even with balancing, the sheer volume of *TinyStories* data may overwhelm the influence of the *BabyLM* data. The model is exposed to a much larger quantity of *TinyStories* content, which could dominate learning and reduce the effectiveness of balancing. Additionally, while the nucleus sampling strategy once again improves performance on the BLiMP Supp. dataset, it does not assist with BLiMP as it did in the `Strict-Small` track.

6 Conclusion

In this work, we explore data augmentation for language pre-training in a limited data setting. Using the *TinyStories* dataset we train GPT-Neo models and probe the relationship between generative ability and dataset size. To measure the effect of augmentation with synthetic data, we train LTG-BERT models on a diverse set of data configurations. Our experiments indicate that while synthesizing high quality data is possible in small data regimes, effectively utilizing it for pre-training can be challenging. Some modest gains are observed in the `Strict` track, while careful balancing shows promise for the `Strict-Small` track. Overall, our evaluation highlights the intricate balance required between data quantity, quality, and integration for effective training. Future work suggests investigation of different data domains, mixtures, and proportions, while precise calibration of hyperparameters may prove critical in exploiting the full benefit of synthetic data in low data pre-training.

7 Limitations

A limitation of our study is the exclusive use of a single LM architecture for both the encoder and decoder components. Our experiments are also limited to specific datasets, employing only *TinyStories* for synthetic data generation and a combination of *TinyStories* and *BabyLM* for encoder training. While these choices are made to ensure experimental control and draw solid conclusions, they limit the generalizability of our results.

Another limitation concerns the creation of the combined dataset. We investigated only a single configuration of the two datasets – including them in equal proportion – and the documents within a dataset were sampled randomly. We posit that more fine control over the mixture of datasets could further enhance the benefits of our data augmentation technique. Additionally, with regard to generation, the prompting strategy and truncation ratio could be more finely calibrated, in order to improve the balance between data quality and redundancy.

By acknowledging these limitations, we aim to encourage further research in this area, focusing on the impact of data augmentation in size constrained and cognitively plausible language pre-training.

Acknowledgments

The research work was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the 3rd Call for HFRI PhD Fellowships (Fellowship Number 5537).

References

Anthropic. 2024. Claude. `https://www.anthropic.com/claude`. Artificial Intelligence Model.

Benjamin Bergner, Andrii Skliar, Amelie Royer, Tijmen Blankevoort, Yuki Asano, and Babak Ehteshami Bejnordi. 2024. Think big, generate quick: Llm-to-slm for fast autoregressive decoding. *arXiv preprint arXiv:2402.16844*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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. *Preprint*, arXiv:2404.06214.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, et al. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*.

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.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using LLMs: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1679–1705, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Martin Gerlach and Francesc Font-Clos. 2018. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Preprint*, arXiv:1812.08092.

Jill Gilkerson, Jeffrey A. Richards, Steven F. Warren, Judith K. Montgomery, Charles R. Greenwood, D. Kimbrough Oller, John H. L. Hansen, and Terrance D. Paul. 2017. Mapping the early language environment using all-day recordings and automated analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

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. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. *Preprint*, arXiv:1904.09751.

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

Jaap Jumelet, Michael Hanna, Marianne De Heer Kloots, Anna Langedijk, Charlotte Pouw, and Oskar Van Der Wal. 2023. Chapgtp, illc's attempt at raising a babylm: Improving data efficiency by automatic task formation. *arXiv preprint arXiv:2310.11282*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Brian MacWhinney. 2014. *The Childes Project*. Psychology Press.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Preprint*, arXiv:2202.04538.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *CoRR*, abs/1912.01703.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

David Samuel. 2023. Mean BERTs make erratic language teachers: the effectiveness of latent bootstrapping in low-resource settings. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 221–237, Singapore. Association for Computational Linguistics.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: BERT meets British National Corpus. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1954–1974, Dubrovnik, Croatia. Association for Computational Linguistics.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

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. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

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

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. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023a. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2301.11796.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell, editors. 2023b. *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Singapore.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2019. Blimp: A benchmark of linguistic minimal pairs for english. *CoRR*, abs/1912.00582.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Chenghao Xiao, G Thomas Hudson, and Noura Al Moubayed. 2023. Towards more human-like language models based on contextualizer pretraining strategy. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 317–326, Singapore. Association for Computational Linguistics.

Wujiang Xu, Zujie Liang, Jiaojiao Han, Xuying Ning, Wenfang Lin, Linxun Chen, Feng Wei, and Yongfeng Zhang. 2024. Slmrec: Empowering small language models for sequential recommendation. *arXiv preprint arXiv:2405.17890*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.

Yian Zhang, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? In *Proceedings* *of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1112–1125, Online. Association for Computational Linguistics.

Zheyu Zhang, Han Yang, Bolei Ma, David Rügamer, and Ercong Nie. 2023. Baby's CoThought: Leveraging large language models for enhanced reasoning in compact models. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 158–170, Singapore. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

A *BabyLM* dataset

Table 5 contains a detailed overview of the *BabyLM* dataset. For our experiments, we preprocess the data using the methodology from Samuel (2023). The text is normalized and cleaned up in order to ensure a unified format. We cast direct speech in double quotes, remove arbitrary and semantically irrelevant tokens and conserve formatting, where necessary, with a special `[PAR]` symbol.

Dataset	Domain	# Words	
		Strict-Small	Strict
CHILDES (MacWhinney, 2014)	Child-directed speech	2.84M	28.90M
British National Corpus (BNC), dialogue portion [1]	Dialogue	0.93M	7.76M
Project Gutenberg (children's stories) (Gerlach and Font-Clos, 2018)	Written English	2.54M	26.37M
OpenSubtitles (Lison and Tiedemann, 2016)	Movie subtitles	2.04M	19.96M
Simple Wikipedia [2]	Written Simple English	1.45M	14.67M
Switchboard Dialog Act Corpus (Stolcke et al., 2000)	Dialogue	0.15M	1.34M
Total		**9.95M**	**99.01M**

Table 5: Contents of the *BabyLM* datasets for the `Strict` and `Strict-Small` tracks, including the domain and word counts. [1]http://www.natcorp.ox.ac.uk/, [2]https://dumps.wikimedia.org/simplewiki/20241001/.

B *TinyStories* - Detailed Evaluation

In order to demonstrate a tangible example of the augmentation process, and provide the opportunity to directly judge the quality of the generations, we include sample generations for all our GPT-Neo models: $\{5\mathrm{M}, 10\mathrm{M}, 25\mathrm{M}, 50\mathrm{M}, 75\mathrm{M}, 100\mathrm{M}, 440\mathrm{M}\ (\mathrm{V2})\}$, as well as the model released by Eldan and Li (2023) – 373M (V1). We sample a story from the training set, truncate it to around 15% to 30% of its length, and ask the models to generate a completion with greedy decoding. The results are shown in Table 6. The 50M words model generation is also illustrated in Figure 1. We notice that even for the smaller models, the quality of the generation remains good in terms of grammar, coherence to the story's beginning, and creativity. Additionally, there is little similarity to the original completion, even though the prompt is taken from the training set of the models.

C Training Hyperparameters

We list all the hyperparameters used to train the GPT-Neo models on subsets of *TinyStories* in Table 7. Where appropriate, parameters that vary per group of models are listed separately. We also list the hyperparameters used to train the LTG-BERT transformer encoder models in Table 8. We follow the hyperparameter and training regime of Samuel et al. (2023), with some modifications, partly to adjust to the scale of our computational resources and infrastructure.

D GLUE Evaluation Hyperparameters

We list the hyperparameters utilized for fine-tuning the LTG-BERT models on GLUE in Table 9, largely also based on Samuel et al. (2023). We find that it's critical for the stability of training to initialize the added classification head according to the original implementation released by the authors. Specifically, we use a truncated normal distribution $\mathcal{N}(\mu, \sigma^2)$ in the range $[a, b]$ where $\mu = 0$, $\sigma = \sqrt{\frac{2.0}{5.0 \cdot d_{model}}}$, $a = -2\sigma$, $b = 2\sigma$ and d_{model} is the hidden size. The bias of the layer is also initialized to zero. Lastly, one notable difference affecting the downstream performance of our models is that we train with *mixed precision*. We use the fp16 implementation provided by the HuggingFace Trainer module, which reduces the numerical precision of certain variables from fp32 to fp16, increasing computational efficiency.

E LLM Evaluation

For the LLM evaluation, we employ Claude-3.5 Sonnet. Our prompt is a modified version of the one introduced in Eldan and Li (2023). We present our prompt in Table 10.

Model	Generated Text
Prompt	One day, a little boy named Tim went to the park. He saw a big tiger. The tiger
Original	was not mean, but very easy to play with. Tim and the tiger played all day. They had lots of fun. Then, something unexpected happened. The tiger started to shake. Tim was scared. He did not know what was going on. But then, the tiger turned into a nice dog. Tim was very surprised. Tim and the dog played together now. They were very happy. The dog was easy to play with too. At the end of the day, Tim went home with his new friend.
5M	was very big and had a long tail. Tim wanted to be friends with the tiger. Tim said, "Hi tiger, I am Tim. Do you want to play with me?" The tiger looked at Tim and said, "Yes, I want to play with you." Tim and the tiger played all day. They ran, jumped, and laughed. The tiger was very happy. But then, the tiger got tired. It wanted to rest. Tim said, "I am tired. Let's rest." The tiger yawned and said, "Okay, Tim. Let's rest." Tim and the tiger sat under a tree. They talked and laughed. The tiger was happy to have a new friend. They both had a fun day at the park.
10M	was very big and had a long tail. Tim was scared, but he wanted to be brave. He walked up to the tiger and said, "Hi, tiger! I am Tim. Do you want to play with me?" The tiger looked at Tim and said, "Yes, I want to play. But I am scared of the big tiger." Tim was not scared. He thought the tiger was nice. Tim and the tiger played together. They ran and jumped. They had a lot of fun. Then, something unexpected happened. The tiger started to change. It became a tiger! The tiger was not a tiger anymore. It was a tiger! Tim was very surprised. The tiger was not a tiger anymore. It was a big tiger! Tim and the tiger became best friends. They played together every day.
25M	was very big and had a long tail. Tim was scared, but he wanted to be brave. Tim said, "Hi, tiger! I am Tim. Do you want to play with me?" The tiger looked at Tim and said, "Yes, I want to play. But I am very hungry. Can you help me find some food?" Tim thought for a moment and said, "Yes, I can help you. Let's look for food together." They looked and looked, but they could not find any food. Then, something unexpected happened. The tiger started to talk! "I am not a tiger, I am a magic tiger. I can make your wishes come true," said the tiger. Tim was very surprised. He wished for a big ice cream cone. The magic tiger made his wish come true. Tim and the magic tiger became good friends and played in the park every day.
50M	was very nice. Tim wanted to play with the tiger. He said, "Hi, tiger! Let's play!" The tiger did not say anything. Tim was sad. He wanted to find a way to make the tiger happy. Tim had an idea. He found a ball and threw it. The tiger saw the ball and started to play with it. Tim was happy. He said, "Now, the tiger is happy too!" They played together all day. The tiger was not sad anymore. Tim and the tiger became best friends.
75M	was very big and had a long tail. Tim was not scared. He wanted to be friends with the tiger. Tim said, "Hi, tiger! My name is Tim. Do you want to play with me?" The tiger looked at Tim and said, "Yes, I want to play. My name is Sam." Tim and Sam played all day. They ran, jumped, and laughed. They had so much fun. Tim was happy to have a new friend. At the end of the day, Tim said, "Bye, Sam! I will come back to play with you again." Sam said, "Bye, Tim! I will wait for you here." Tim went home and told his mom about his new friend.
100M	was sad. Tim asked, "Why are you sad, Mr. Tiger?" The tiger said, "I lost my toy. Can you help me find it?" Tim wanted to help the tiger. They looked for the toy together. They looked under the trees and behind the bushes. Then, they saw a big box. The tiger said, "I think my toy is in the box." Tim opened the box. Inside, there was a big cake! The tiger was happy. But, the cake was not a real tiger. It was a toy tiger! Tim laughed and said, "Silly tiger, you are not a real tiger!" They played with the toy tiger and had a fun day at the park.
440M (V2)	was very hungry. Tim wanted to help the tiger. He thought, "I will feed the tiger some food." Tim went home and got some food. He came back to the park and gave the food to the tiger. The tiger ate the food and was very happy. Tim felt good because he helped the tiger. The tiger said, "Thank you, Tim. You are a good friend." Tim and the tiger played in the park all day. They had lots of fun. Tim learned that it is good to help others. The tiger learned that it is good to be kind and help others. And they both lived happily ever after.
373M (V1)	was sad. Tim wanted to help the tiger. He thought the tiger needed a friend. Tim went to the tiger and said, "Hi, I am Tim. I want to help you." The tiger looked at Tim and said, "Thank you, Tim. I am sad because I am alone. I need a friend." Tim thought for a moment and said, "I will be your friend, tiger. We can play together." The tiger was happy. Tim and the tiger played all day. They had lots of fun. The moral of the story is: Helping others can make you happy too.

Table 6: Story completions for models trained on different amounts of *TinyStories* data. The original prompt and the training set completion are also included.

Hyperparameter	GPT-Neo – 5 / 10 / 25 / 50, 75, 100, 440 (M)
Number of Parameters	41M
Number of Layers	4
Attention Heads	16
Hidden size	768
Layer norm ϵ	1.0e-5
Sequence Length	512
Max position embeddings	512
Attention Dropout	0.50 / 0.40 / 0.25 / 0.20
Classifier Dropout	0.50 / 0.40 / 0.25 / 0.20
Embed. Dropout	0.50 / 0.40 / 0.25 / 0.20
Resid. Dropout	0.50 / 0.40 / 0.25 / 0.20
Summary first Dropout	0.40 / 0.30 / 0.15 / 0.10
Weight decay	0.20 / 0.20 / 0.20 / 0.10
Vocab Size	6411 / 6411 / 16384 / 16384
Context length	512
batch size	24
gradient accumulation steps	32
gradient clipping	2.0
Training steps	15 000
optimizer	AdamW
Adam β_1	0.9
Adam β_2	0.95
Adam ϵ	1.0e-8
Initial learning rate	5.0e-4
Final learning rate	5.0e-5
Learning rate scheduler schedule	cosine
Warmup ratio	1.6%

Table 7: Hyperparameters used for training GPT-Neo models on *TinyStories*.

Hyperparameter	Strict	Strict-Small
Number of parameters	98M	24M
Number of layers	12	12
Attention heads	12	6
Hidden size	768	384
FF intermediate size	2048	1024
Position Bucket size	32	32
Layer norm ϵ	1e-7	1e-7
Vocabulary size	16 384	6 144
Sequence length	128	128
Max position embeddings	512	512
Hidden dropout	0.1	0.1
Attention dropout	0.1	0.1
Training steps	20 000	20 000
Batch size	80	80
Gradient Accumulation Steps	32	32
Warmup ratio	1.6%	1.6%
Initial learning rate	6e-4	6e-4
Final learning rate	6e-5	6e-5
Learning rate scheduler	cosine	cosine
Weight decay	0.1	0.1
Optimizer	AdamW	AdamW
Adam ϵ	1e-6	1e-6
Adam β_1	0.9	0.9
Adam β_2	0.98	0.98
Gradient clipping	2.0	2.0

Table 8: Hyperparameters used to train all LTG-BERT models for the different tracks. With *max position embedding* we refer to the architectural capacity of the model – the model was trained with *sequence length = 128*.

Hyperparameter	BoolQ, MNLI, MNLI-mm, MRPC MultiRC, QNLI, QQP, SST-2	CoLA, RTE, WSC
Batch size	32	16
Number of epochs	10	10
Dropot	0.1	0.1
Warmup proportion	0.1	0.1
Learning Rate	3.0e-5	3.0e-5
Decay type	linear	linear
Weight decay	0.01	0.01
Optimizer	AdamW	AdamW
Adam ϵ	1.0e-6	1.0e-6
Adam β_1	0.9	0.9
Adam β_2	0.999	0.999
Max length	128	128
Gradient clip	2.0	2.0
Mixed precision	True	True

Table 9: Hyperparameters used for finetuning the LTG-BERT models on GLUE.

In the following exercise, the student is given a beginning of a story. The student needs to complete it into a full story. The exercise tests the student's language abilities and creativity. The symbol *** marks the separator between the prescribed beginning and the student's completion:

Once upon a time, there was tiger who liked to play the guitar. One day, a bunny heard the guitar from a distance and *** asked, "Hello tiger, can you play the guitar for me?" The tiger said, "Yes! I can!" So the bunny plucked up the guitar and started to play. But then, the tiger roared and kicked the guitar too hard. The guitar made a big mess. The panda was so sad that he hopped away and never came back. The end.

Please provide your general assessment about the part written by the student (the one after the *** symbol). Is it gramatically correct? Is it consistent with the beginning of the story? Pay special attention to whether the student manages to complete the sentence which is split in the middle by the separator ***.

I'll provide an assessment of the student's completion of the story:
Grammar: The student's portion is generally grammatically correct, with a few minor issues:
...
...
In summary, while the student completes the split sentence correctly, the overall story lacks coherence and consistency with the given beginning.

Now, grade the student's completion in terms of 1. Grammar, 2. Creativity, 3. Consistency with the story's beginning and whether the plot makes sense. Please provide grades from a scale of 1-10 for each of the requested categories, namely: 1. Grammar, 2. Creativity, 3. Consistency. Moreover, please provide your best guess of what the age of the student might be, as reflected from the completion. Choose from possible age groups: A: 3 or under. B: 4-5. C: 6-7. D: 8-9. E: 10-12. F: 13-16.

Here are the grades for the student's completion on a scale of 1-10:

1. Grammar: 7/10 ... 2. Creativity: 5/10 ... Consistency: 3/10 ...

Age group estimate: B: 4-5 years old ...

Table 10: A sample interaction with Claude-3.5 Sonnet. The story is generated by the GPT-Neo model trained on 50M words of *TinyStories*. We include the prompt which also contains the model's generation. The LLM's responses is stylized in *italic*. For brevity, part of the LLM response was omitted and replaced with '...'.

AntLM: Bridging Causal and Masked Language Models

Xinru Yu[1*], **Bin Guo**[1*], **Shiwei Luo**[3*], **Jie Wang**[1], **Tao Ji**[2†], **Yuanbin Wu**[1†]
[1] School of Computer Science and Technology, East China Normal University
[2] School of Computer Science, Fudan University
[3] School of Computer Science and Technology, Harbin Engineering University
{xryu@stu,binguo@stu,jiewang@stu,ybwu@cs}.ecnu.edu.cn, taoji@fudan.edu.cn, shiweiluomo@gmail.com

Abstract

Causal Language Modeling (CLM) and Masked Language Modeling (MLM) are two mainstream learning paradigms based on Transformer networks, specifically the Decoder-only and Encoder-only architectures. The strengths of each paradigm in downstream tasks have shown a mix of advantages and disadvantages. In the past BabyLM Challenge 2023, although the MLM paradigm achieved the best average performance, the CLM paradigm demonstrated significantly faster convergence rates. For the BabyLM Challenge 2024, we propose a novel language modeling paradigm named **AntLM**, which integrates both CLM and MLM to leverage the advantages of these two classic paradigms. We chose the strict-small track and conducted experiments on two foundation models: BabyLlama, representing CLM, and LTG-BERT, representing MLM. During the training process for specific foundation models, we alternate between applying CLM or MLM training objectives and causal or bidirectional attention masks. Experimental results show that combining the two pretraining objectives leverages their strengths, enhancing overall training performance. Under the same epochs, $\text{AntLM}_{\text{BabyLlama}}$ improves Macro-average by 1%, and $\text{AntLM}_{\text{LTG-BERT}}$ achieves a 2.2% increase over the baselines.

1 Introduction

Language Modeling (LM) is a core task in NLP and a key technology for natural language understanding and generation, supporting a wide range of applications including machine translation (Hendy et al., 2023), speech recognition (Prabhavalkar et al., 2023), sentiment analysis (Tan et al., 2023), and information extraction (Wei et al., 2023). Over the past decades, LM has seen significant development, evolving from simple models like n-grams (Suen, 1979) to more sophisticated models, such as recurrent neural networks (Elman, 1990), long short-term memory networks (Hochreiter, 1997), and more recently, Transformer-based large language models (LLMs) like GPT (Radford et al., 2019) and BERT (Devlin, 2018). LLMs have demonstrated human-like or even superhuman performance in language modeling.

However, the tremendous success of LLMs relies on learning from massive corpora, which is not as data-efficient and low-energy as human language learning. The BabyLM Challenge 2023 (Warstadt et al., 2023a) and 2024 (Choshen et al., 2024) is a shared task over two consecutive years. It aims to encourage the discovery of more effective methods for training models using limited data. Considering that a 13-year-old child has encountered fewer than 100 million words in their lifetime, the shared task has introduced the *strict-small track*[1]. These tracks confine pre-training data to 10 million and 100 million words. These datasets consist of child-accessible materials, such as books, conversations, and Wikipedia entries, to enhance the relevance of language model pre-training to human language learning processes. Compared to 2023, the 2024 competition removed the Children's Book Test (Hill et al., 2016) and QCRI Educational Domain Corpus datasets (Abdelali et al., 2014). The 2024 competition also reduced the proportion of OpenSubtitles (Lison and Tiedemann, 2016) dataset while increasing the proportions of CHILDES (MacWhinney, 2000) and Project Gutenberg (Gerlach and Font-Clos, 2020) datasets.

The current investigation of LMs primarily adopts two predominant modeling paradigms: Causal Language Models (CLMs), represented by GPT (Radford et al., 2019), and Masked Language Models (MLMs), represented by BERT (Devlin,

* Equal contribution.
† Corresponding authors.

[1] Due to limitations in computational resources, we have not yet explored the *strict track* and the *multimodal track*.

The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, pages 324–331
November 15-16, 2024 ©2024 Association for Computational Linguistics

2018). CLMs employ next-token prediction as their training objective, which is predicting the next token given the preceding context, and they perform exceptionally well on generative tasks. On the other hand, the training objective of MLM is the random selection and masking of some tokens in the input text, following which the model is trained to predict the original unmasked tokens. Due to its global information modeling capabilities, this approach excels in tasks necessitating the capture of bidirectional contextual information, such as text classification. Considering these modeling paradigms' strengths, this paper raises an important question: Could the two modeling methodologies be seamlessly integrated?

Intuitively, performing the MLM task allows the model to learn bidirectional contextual encoding of text, while the CLM task enables the model to predict and generate text based on prior content. These two learning objectives are not in conflict and could potentially be integrated. Analogous to a child learning a new language via practicing both cloze exercises and writing assignments, the training mechanism for a model can similarly employ a multi-task strategy. Therefore, we consider enabling our model to learn both tasks concurrently. To achieve this, we adopt a unified model architecture and alternate the training objective between MLM and CLM tasks. This approach attempts to mimic the human learning process, hence helping the model acquire deeper knowledge from a limited amount of text data.

To examine the effect of integrating MLM and CLM pretraining tasks on model performance, we conducted experiments using LTG-BERT and BabyLlama[2] as base models, testing on the BabyLM2024 10M datasets. LTG-BERT, an Encoder-only model, and BabyLlama, a Decoder-only model, are notable architectures from the 2023 BabyLM Challenge .The results indicate that both LTG-BERT and BabyLlama showed improvements in macroaverage scores. These experiments confirm that the integration of these two pretraining objectives can positively impact model training.

2 Related Work

Causal Language Models have played a pivotal role in the development of NLP, particularly in tasks involving sequence generation. The foundational work by OpenAI on the Generative Pre-trained Transformer (GPT) (Radford, 2018) marked a significant breakthrough in the use of CLMs for a variety of NLP applications. GPT (Radford, 2018) models the probability of each token in a sequence based on all preceding tokens, enabling it to perform well on tasks like text completion, machine translation, and summarization. The subsequent release of GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) further illustrated the power of scaling CLMs. These models, with their increased parameter sizes and training data, have set new performance benchmarks in tasks like zero-shot and few-shot learning. The GPT family firmly established the dominance of autoregressive models in generative tasks. More recently, Meta introduced the LLaMA (Touvron et al., 2023) series, which demonstrated that highly capable CLMs could be trained efficiently on fewer parameters and less compute than earlier models like GPT-3. LLaMA, designed to be accessible for academic research, retains the autoregressive framework while achieving competitive performance across a range of NLP tasks.

Masked Language Model is a training approach used to develop deep bidirectional representations of context, often referred to as a cloze task (Taylor, 1953). Specifically, a special token [MASK] is employed to randomly mask a proportion of input tokens, and the model is trained to predict these masked tokens. This training task was first innovatively introduced in BERT (Devlin, 2018) and has been adopted in subsequent models like RoBERTa (Liu, 2019) and ALBERT (Lan, 2019). Research has also led to improvements in MLM tasks, such as in SpanBERT (Joshi et al., 2020), where the model is trained to predict spans of words instead of individual tokens, enhancing its ability to capture long-range dependencies.

Unified modeling refers to using a single model architecture to handle multiple training and evaluation tasks. In the T5 (Raffel et al., 2020) model, various downstream tasks were reformulated as text-to-text tasks, significantly enhancing the model's ability for multitask learning. Moreover, many related works (Sanh et al., 2019; Liu et al., 2020) have also applied unified modeling for multitask training and evaluation, making it a common approach to improve the generalization ability of models. UniLM (Dong et al., 2019), based on the BERT architecture, is one of the significant endeavors in unified modeling. By employing specific self-attention

[2]We only utilized the BabyLlama architecture and did not apply the knowledge distillation method here.

masks, UniLM controls the contextual information used during prediction. When predicting tokens, it not only trains like an autoencoding language model by leveraging the context of masked tokens but also performs left-to-right training like an autoregressive language model. Additionally, UniLM can function similarly to encoder-decoder architectures by encoding the first input text and then generating sequences from left to right. By switching the attention matrix, it seamlessly transitions between different training tasks and downstream application scenarios.

Existing methods have unified CLM and MLM networks regarding model architecture and parameter sharing. However, research on unifying their training objectives remains unexplored. This paper is the first to bridge the two classic training objectives.

3 Methods

3.1 Preliminaries

BabyLlama (Timiryasov and Tastet, 2023) was proved to be effective in BabyLM2023 and is included as one of the baselines officially provided by BabyLM2024. BabyLlama (Timiryasov and Tastet, 2023) employed knowledge distillation, transferring the knowledge from two teacher models — a GPT-2 model with 705 million parameters and a LLaMA model with 360 million parameters — into a compact BabyLlama "student" model with just 58 million parameters. Given that our own replication of the BabyLlama model through distillation did not achieve ideal results, we opted to use only the BabyLlama architecture with a parameter size of 97 million. The BabyLlama model employs the classic CLM paradigm (Radford, 2018), where given the first n tokens in a sequence, the model predicts the token at position $n+1$. The next-token prediction (NTP) training objective is to minimize the negative log-likelihood loss of predicting the next token at each timestep. To achieve this, a causal mask is applied in the self-attention mechanism. This mask is represented as a lower triangular matrix, ensuring each token can only attend to its preceding tokens. Formally, for an input sequence of length T, $x_1, x_2, \ldots, x_T$, the corresponding attention mask M is a $T \times T$ lower triangular matrix, where M_{ij} indicates whether the token at position i should attend to the token at position j This masking strategy effectively prevents the model from accessing future information during training, thereby capturing the sequential order and dependencies within the data.

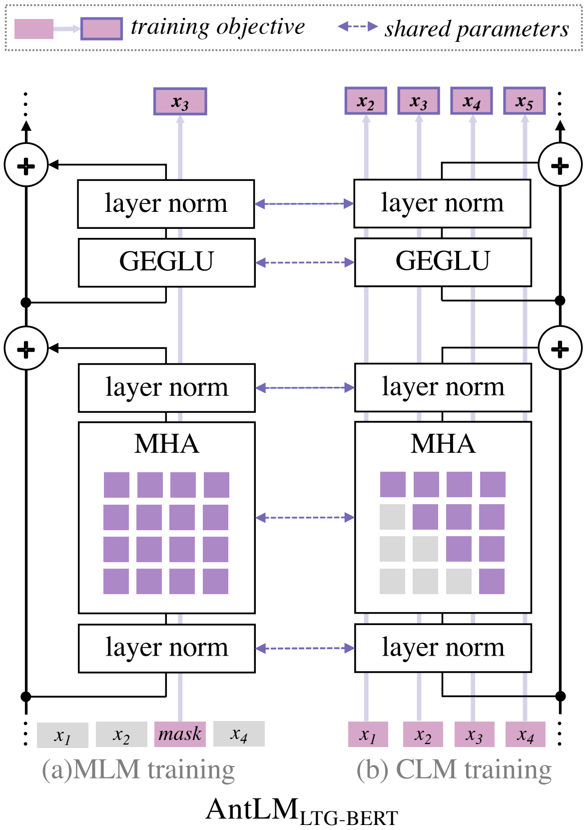


Figure 1: A diagram of AntLM$_{\text{LTG-BERT}}$. Based on the LTG-BERT architecture, we propose a joint MLM and CLM training objective. It is worth noting that the two objectives fully share parameters, but differ in their attention masks. The diagram also applies to AntLM$_{\text{BabyLlama}}$, with the difference in the architecture (e.g., positional encoding and the activation function of GLU).

In BabyLM2023 (Warstadt et al., 2023b), experiments with Boot-BERT (Samuel, 2023) and ELC-BERT (Charpentier and Samuel, 2023) demonstrated the effectiveness of the LTG-BERT (Samuel et al., 2023) architecture. LTG-BERT is also one of the official baselines in BabyLM2024. The LTG-BERT model incorporates several key architectural improvements, including NormFormer layer normalization (Shleifer et al., 2021), disentangled attention with relative position embeddings (He et al., 2020), and gated-linear activation function (Shazeer, 2020). The training objective of LTG-BERT is self-supervised Masked Language Modeling (MLM). During training, 15% of the tokens in the input sequence are randomly selected for replacement. Of these, 80% are masked, 10% are substituted with random tokens, and the remaining 10% are unchanged. The model is then trained to predict the original masked tokens based on the context. LTG-BERT explores three common masking strategies: subwords, whole words, and spans. Experimental results indicate that span-based masking yields slightly better performance compared to the other methods.

3.2 Our Approach

Inspired by the way children learn languages through both cloze exercises and writing assignments, our work constructs a unified training framework that integrates CLM and MLM. In this unified framework, we switch between the two training paradigms alternately. CLM uses a causal mask to enforce sequential dependencies and MLM employs bidirectional attention, enabling the model to predict masked tokens by leveraging both preceding and succeeding context. By combining these two training objectives, the model not only excels at autoregressive tasks like text generation but also achieves a deeper semantic understanding of language by capturing broader contextual information through bidirectional attention.

In our approach, we integrate CLM and MLM by alternating between these training objectives during the pre-training phase. After training the model on one objective for a specified number of epochs, we switch to the other objective. The switch between training objectives is implemented by modifying the model's input and attention matrix. For the MLM task, 15% of the tokens in the input are randomly selected and replaced. The model utilizes bidirectional attention to predict the original tokens based on the surrounding context. In contrast, for the CLM task, no token replacement is required in the input. The model employs causal attention to predict the next token based on the preceding tokens.

4 Experiment

Data Preprocessing For the data preprocessing part, we adopt the data handling procedures from the BootBERT (Samuel, 2023) method, which performed well in the previous round of BabyLM Challenge. Preprocessing includes steps like normalizing punctuation, reconstructing sentence structures, and removing duplicate text. These preprocessing steps help ensure cleaner and more structured input data, contributing to better model performance.

Name	BabyLlama	LTG-BERT
layers	12	12
attention heads	12	12
hidden size	768	768
intermediate size	2048	2048
vocabulary size	16k	16k
position bucket	–	32

Table 1: Model Hyper-parameters.

Baselines We adopt the official baseline provided by the BabyLM Challenge as our benchmark, using the results achieved by the best-performing models from the previous round, namely LTG-BERT and BabyLlama, see Table 2.

Experiment Settings In our experiments, we used both the BabyLlama and LGT-BERT models to evaluate the performance of a hybrid training strategy combining Causal Language Modeling (CLM) and Masked Language Modeling (MLM). For both model architectures, we used the same set of parameters, as shown in the table 1 and optimized the training process using the AdamW optimizer. Additionally, we employed the bfloat16 data type to enhance computational efficiency. For the BabyLlama model, we used a batch size of 512 with an initial learning rate set to 7×10^{-4}. The learning rate scheduler followed a cosine decay during the CLM training phase and a cosine with restarts scheduler during the MLM phase, with the number of cycles set to every four epochs . For the LGT-BERT model, we employed a batch size of 1024, with an initial learning rate of 5×10^{-4}. In all training phases, we used a cosine with restarts scheduler, with the num cycles set to 4. Our hyperparameters were determined through multiple experiments, building upon the hyperparameter settings from the previous works (Timiryasov and Tastet, 2023; Samuel et al., 2023) to find the optimal values. The training process alternated between CLM and MLM objectives over multiple epochs. We used the notation "x_CLM + y_MLM..." to indicate that, *in sequential order*, x epochs are trained in the CLM training mode, followed by y epochs in the MLM training mode, and so on.

4.1 Main Results

In this section, we evaluate the performance of BabyLlama and LTG-BERT across multiple bench-

Model	Data	BLiMP	BLiMP Supplement	EWoK	GLUE	*Macro average*
BabyLlama†	10M	**69.8**	59.5	50.7	63.3	60.8
BabyLlama	10M	68.1	60.4	50.4	65.5	61.1
AntLM$_{\text{BabyLlama}}$	10M	69.4	**60.7**	**51.1**	**67.4**	**62.1**
BabyLlama†	100M	73.1	60.6	**52.1**	**69.0**	63.7
LTG-BERT†	100M	69.2	**66.5**	51.9	68.4	64.0
BabyLlama	100M	**74.9**	66.0	52.0	66.3	**64.8**
LTG-BERT†	10M	60.6	60.8	48.9	60.3	57.5
LTG-BERT	10M	62.6	**65.4**	62.3	64.9	63.8
AntLM$_{\text{LTG-BERT}}$	10M	**72.3**	62.6	**63.0**	**66.0**	**66.0**

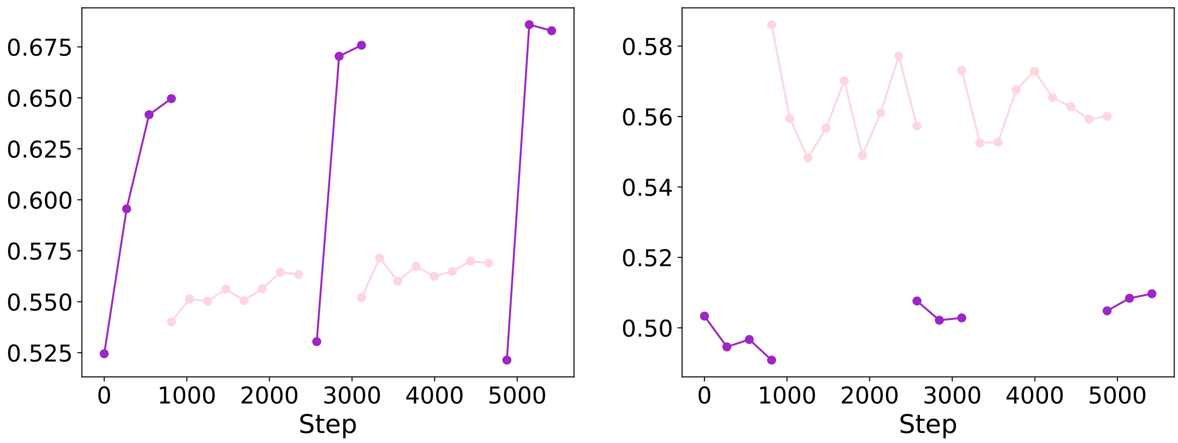
Table 2: Main experimental results. The † indicates results from the official report. The official BabyLlama leverages knowledge distillation, while our AntLM$_{\text{BabyLlama}}$ is based solely on the architecture of BabyLlama without knowledge distillation methods. Due to limitations in time and resources, we have not attempted AntLM on the 100M track, this will be part of our future work.

Training Stage	BLiMP	BLiMP Supplement	EWoK	Avg.
AntLM$_{\text{BabyLlama}}$				
8	68.2	56.7	50.5	58.5
16	56.8	58.4	57.2	57.5
24	68.1	60.4	50.4	59.6
24	56.9	57.8	**58.3**	57.7
4 16 4	**69.4**	**60.7**	51.1	**60.4**
AntLM$_{\text{LTG-BERT}}$				
12	69.9	56.4	50.8	59.0
60	62.8	**63.5**	64.2	63.5
72	70.0	57.2	51.9	57.9
72	69.4	61.1	**64.5**	65.0
6 60 6	**72.3**	62.5	63.0	**66.0**

Table 3: The effect of integrating CLM and MLM training objectives on BabyLlama and LTG-BERT.

marks, including BLiMP, BLiMP Supplement, EWoK, and GLUE. Our experiments primarily focus on assessing the impact of integrating CLM and MLM training objectives on the overall results, comparing the baseline performance of both BabyLlama and LTG-BERT with the configurations we propose.

As shown in Table 2, our models with integrated training objectives consistently outperform the official baseline scores on both the LTG-BERT and BabyLlama models. Notably, the improvements on LTG-BERT are particularly significant, demonstrating the effectiveness of our approach. To further validate the effectiveness of alternating training objectives CLM and MLM, we conducted an in-depth experiment with the BabyLlama model. Given the lengthy training times associated with the GLUE dataset, we opted to evaluate our results on the BLiMP, BLiMP Supplement, and EWoK datasets. As shown in Table 3, the model trained with the *4_CLM+16_MLM+4_CLM* strategy significantly outperformed those trained solely with *8_CLM* or *16_MLM*. This finding indicates that combining these two training objectives enables the model to simultaneously acquire bidirectional context understanding and sequence generation capabilities. Under the same training epochs, the *4_CLM+16_CLM+4_CLM* combination demonstrated clear advantages over the pure *24_CLM* and *24_MLM* models, further confirming that the inte-



(a) BLiMP (b) EWok

Figure 2: The phased experimental results on three datasets. The evaluation line chart for each stage of "3_ CLM + 8_ MLM + 2_ CLM + 8_ MLM + 3_ CLM" on the BabyLlama model. The reason for the discontinuity in evaluation results between training phases is that we applied the evaluation method corresponding to the specific task categories at each stage of the training process.

gration of these two training objectives is crucial for achieving optimal performance, highlighting the complementary relationship between CLM and MLM. We also conducted similar experiments on the LTG-BERT, the results are shown on same Table.

Additionally, we explored the performance of these training modes across different datasets. As shown in Figure 2, MLM performs significantly better on the EWoK dataset, while CLM exhibits more pronounced and sensitive results on the BLiMP dataset. This indicates that different training approaches have varying impacts on distinct datasets. Thus, the integrated experiments that combine both training methods can better leverage their strengths and enhance overall performance.

4.2 Ablation Study

To investigate the effects of various factors on the evaluation task results within the integrated experiments, we conducted ablation studies focusing on two variables: alternating frequency and alternating order. In the BabyLlama model, we maintained a constant total number of training epochs at 24 (8 epochs for the CLM phase and 16 epochs for the MLM phase). Specifically, for the alternating order, we adjusted the alternating sequence of training between the CLM and MLM phases while keeping the overall epoch count unchanged. For alternating frequency, we divided the training process into more frequent alternating stages. The experimental results, as shown in Table 4, indicate that variations in these two factors do not lead to significant declines in evaluation outcomes, suggesting that our approach is stable. We hypothesize that the decrease in performance with an increased frequency of alternations may be attributed to smaller epoch sizes in each training phase, which could hinder convergence on the respective tasks.

Furthermore, we found that the best performance was achieved when the CLM training phase was placed at both the beginning and the end of the training sequence, which could be due to the greater impact of CLM compared to MLM. Although CLM does not inherently have a higher performance ceiling (as last year's winner was an MLM-based model), but it converges more rapidly. CLM performs sequential prediction training on every token, while MLM focuses only on masked tokens. Thus, we suggest that CLM captures more learning within a single epoch than MLM.

5 Conclusion

In this study, we propose AntLM, a model that applies to multiple natural language-related tasks in the BabyLM Challenge by alternating between Causal Language Modeling (CLM) and Masked Language Modeling (MLM) during training. Experimental results demonstrate that AntLM achieves either superior or comparable performance to the baseline across all evaluation tasks.

Additionally, we found that CLM and MLM have different impacts on various evaluation tasks, suggesting that these training tasks guide the model

Training Stage	BLiMP	BLiMP Supplement	EWoK	Avg.
AntLM$_{\text{BabyLlama}}$				
8 16	68.2	56.7	50.5	58.5
16 8	68.4	**61.1**	50.1	59.9
4 16 4	**69.4**	60.7	**51.1**	**60.4**
8 8 8	67.2	59.2	50.2	58.9
4 8 4 8	68.8	60.6	50.7	60.0
8 4 8 4	68.6	59.1	51.0	59.6
3 8 2 8 3	69.3	60.1	50.8	60.1
1 3 1 3 1 2 1 2 1 2 1 2 1 2 1	67.3	55.2	50.4	57.6

Table 4: The effect of alternating frequency (low or high) and alternating order of CLM and MLM training objectives on BabyLlama. All were trained for a total of 24 epochs.

to learn distinct aspects of human language. We believe this difference is the key reason why integrated training yields effective results, as the model benefits from the knowledge learned from both training approaches. This finding also raises an intriguing question: do different training tasks allow models to capture only specific portions of natural language knowledge? Due to resource limitations, we were unable to explore additional ideas and approaches in this study. In future work, we plan to address these limitations by expanding our resources and support, allowing us to further investigate these potential directions.

Moreover, we conducted experiments with varying numbers and sequences of alternating training, and the results suggest that specific integrated training methods are more effective in achieving optimal evaluation outcomes.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The amara corpus: Building parallel language resources for the educational domain. In *LREC*, volume 14, pages 1044–1054.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts bert. *arXiv preprint arXiv:2311.02265*.

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. *Preprint*, arXiv:2404.06214.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *Advances in neural information processing systems*, 32.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The Goldilocks principle: Reading children's books with explicit memory representations. *Preprint*, arXiv:1511.02301.

S Hochreiter. 1997. Long short-term memory. *Neural Computation MIT-Press*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8:64–77.

Z Lan. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2020. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8433–8440.

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Rohit Prabhavalkar, Takaaki Hori, Tara N Sainath, Ralf Schlüter, and Shinji Watanabe. 2023. End-to-end speech recognition: A survey. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Alec Radford. 2018. Improving language understanding by generative pre-training.

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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

David Samuel. 2023. Mean berts make erratic language teachers: the effectiveness of latent bootstrapping in low-resource settings. *arXiv preprint arXiv:2310.19420*.

David Samuel, Andrey Kutuzov, Lilja Øvrelid, and Erik Velldal. 2023. Trained on 100 million words and still in shape: Bert meets british national corpus. *arXiv preprint arXiv:2303.09859*.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6949–6956.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Sam Shleifer, Jason Weston, and Myle Ott. 2021. Normformer: Improved transformer pretraining with extra normalization. *arXiv preprint arXiv:2110.09456*.

Ching Y. Suen. 1979. n-gram statistics for natural language understanding and text processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):164–172.

Kian Long Tan, Chin Poo Lee, and Kian Ming Lim. 2023. A survey of sentiment analysis: Approaches, datasets, and future research. *Applied Sciences*, 13(7):4550.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.

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. *Preprint*, arXiv:2308.02019.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023a. Call for papers – the babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2301.11796.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell, editors. 2023b. *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Singapore.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Author Index

Akbik, Alan, 82
AlKhamissi, Badr, 244

Barbini, Matilde, 106
Bartelt, Christian, 118
Behr, Rufus, 140
Beinborn, Lisa, 37
Berend, Gábor, 159
Bianchessi, Maria Letizia Piccini, 106
Bisazza, Arianna, 252
Bondielli, Alessandro, 189
Bressan, Veronica, 106
Bunzeck, Bastian, 54
Buttery, Paula, 37, 174
Bylinina, Lisa, 166

Caines, Andrew, 37
Capone, Luca, 189
Charpentier, Lucas Georges Gabriel, 262
Chesi, Cristiano, 106
Chi, Jou-An, 147
Choshen, Leshem, 1
Cotterell, Ryan, 1

DeBenedetto, Justin, 212
Diehl Martinez, Richard, 37, 174
Dousti, Mohammad Javad, 22
Duran, Daniel, 54

Edman, Lukas, 166

Fahim, Abrar, 65
Filandrianos, Giorgos, 308
Fraser, Alexander, 166
Fukatsu, Akiyo, 252
Fusco, Achille, 106
Fyshe, Alona, 65

Gaines, Dylan, 221
Ghanizadeh, Mohammad Amin, 22
Gharaee, Ali, 28
Ghorbanpour, Faeze, 166
Gokce, Abdulkadir, 244
Golde, Jonas, 82
Goriely, Zebulon, 37, 174
Guo, Bin, 324

Haga, Akari, 252
Haller, Patrick, 82
Hancharova, Alina, 28
Havens, Timothy, 221
Hepner, Bryce, 229
Hong, Xudong, 237
Hsieh, Shu-Kai, 147
Hu, Michael, 1

Iyer, Srikrishna, 197

Ji, Tao, 324

Klerings, Alina, 118
Kosireddy, Tagore Rao, 221
Kumar, Mayank, 28

Lenci, Alessandro, 189
Li, Kevin, 221
Linzen, Tal, 1
Loáiciga, Sharid, 237
Lucas, Evan, 221
Luo, Shiwei, 324
Lyberatos, Vassilis, 308
Lyman, Alex Mark, 229
Lymperaiou, Maria, 308

Matusevych, Yevgen, 95
Mehrer, Johannes, 244
Mueller, Aaron, 1, 118
Murphy, Alex, 65

Nair, Aakarsh, 28
Neri, Sofia, 106
Nguyen, Hiep, 212
Nissim, Malvina, 95

Oba, Miyu, 252
Oseki, Yohei, 252

Paek, Nathan, 284, 302
Prevot, Laurent, 147

Ross, Candace, 1
Rossi, Sarah, 106

Saha, Rohan, 65
Salhan, Suchir, 174
Samuel, David, 262

Sayeed, Asad B., 237
Schade, Leonie, 54
Schrimpf, Martin, 244
Sgrizzi, Tommaso, 106
Shaozhen, Shi, 95
Stamou, Giorgos, 308

Tang, Yingtian, 244
Tastet, Jean-Loup, 292
Theodoropoulos, Nikitas, 308
Timiryasov, Inar, 292

Wang, Jie, 324
Wang, Sheng-Fu, 147
Warstadt, Alex, 1
Wilcox, Ethan Gotlieb, 1
Williams, Adina, 1
Wu, Yuanbin, 324

Yam, Hong Meng, 284, 302
Yip, Lynn, 212
Yu, Xinru, 324

Zarrieß, Sina, 54
Zhuang, Chengxu, 1