Bigger is not always better: The importance of human-scale language modeling for psycholinguistics

Ethan Wilcox[a,1], Michael Y. Hu[b], Aaron Mueller[c], Alex Warstadt[a], Leshem Choshen[d], Chengxu Zhuang[d], Adina Williams[e,2], Ryan Cotterell[a,2], Tal Linzen[b,2]

[a]*Department of Computer Science, ETH Zürich, Universitätstrasse 6, 8092, Zürich, Switzerland*
[b]*Center for Data Science, New York University, 10 Washington Place, New York, NY, 10003, USA*
[c]*Khoury College of Computer Sciences, Northeastern University, 440 Huntington Avenue, Boston, MA, 02115, USA*
[d]*Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA, 02139, USA*
[e]*FAIR Laboratories, Meta Platforms Inc., 390 9th Ave, New York, NY, 10001, USA*

Abstract

Neural network language models can learn a surprising amount about language by predicting upcoming words in a corpus. Recent language technologies work has demonstrated that large performance improvements can arise from simply increasing ("scaling") the size of the data sets they are trained on (and, correspondingly, the number of parameters in those models); accordingly, many contemporary systems are trained on trillions of words. While largely beneficial to performance on language applications, scaling has several downsides for both computational psycholinguistics and natural language processing research. We discuss the scientific challenges presented by scaling, as well as the benefits that would result from human-scale language modeling research. In the second half of this paper, we report on takeaways from two efforts to bring about human-scale language model pretraining. First, we report on the first iteration of the BabyLM Challenge, a shared task organized by the authors that asked participants to train a language model on 100 million words or less. Second, we present experiments to answer open questions from the findings of the BabyLM Challenge: namely, are a significant amount of computational resources required to achieve high performance, even at such small scales? We find that high performance can be achieved at small data scales and with typical academic-scale computational resources.

[1]To whom correspondence should be addressed. E-mail: ethan.wilcox@inf.ethz.ch
[2]Senior authors.

Keywords: Language modeling, Scaling, Cognitive modeling, Connectionist networks, Psycholinguistics, Language acquisition

1. Introduction

Connectionist modeling has been a core theoretical and empirical tool for psycholinguistics research over the past three decades. It provides a key theoretical paradigm for modeling how symbolic representations can arise in a distributed system. Furthermore, testing the behaviors of connectionist models against human data allows researchers to evaluate a variety of psycholinguistic theories, especially in the areas of language learning and language processing. The focus of this article will be on connectionist, or neural-network-based, **language models** (LMs). LMs are data structures that predict the probability of a string of text. Generally, they learn to model the distribution of units of text given some training dataset. (The units of text are referred to as **tokens**.) In recent years, language modeling has seen a surge in interest and popularity, due largely to advances in the underlying technological methods, and their resulting generative AI technologies. These advances have been driven largely by **scaling** (Kaplan et al., 2020)—i.e., increasing the number of parameters in a language model, training it on larger and larger amounts of data, or often both. This article takes a critical look at scaling, and at the massive training dataset sizes associated with it, from a psycholinguistics perspective. We argue that bigger is not always better, and that future success in connectionist modeling of psycholinguistic processes will require balancing the training and analysis of large models alongside models that are more humanlike with respect to the scale of the training dataset.

First, we discuss the impact of scaling on the important, yet often "frictional" (Pater, 2019) relationship between connectionist, deep-learning modeling and linguistics. We outline two key ways in which connectionist modeling can contribute to linguistics research, focusing on the role that can be played by **language models**, which are algorithms that assign a probability to a string of text. We argue that the utility of language models for psycholinguistics research is jeopardized by the current trend toward larger and more data-intensive models. We propose that these downsides can be mitigated by devoting effort toward building more data-efficient connectionist models that are trained on more developmentally plausible datasets in terms of size, genre, and input modality.

Second, we discuss the impact of scaling on machine learning (ML) and natural language processing (NLP) research. Although generally beneficial for NLP

applications, we argue that the scaling trend is not without its downsides, and highlight three: First, the focus on evaluating language models based solely on performance incentivizes scaling, at the cost of not incentivizing research into data-efficient models (Linzen, 2020). More data-efficient models are essential for cases of true data scarcity, for example, with proprietary datasets or for creating language technologies for low-resource languages. Second, smaller datasets are easier to curate and control for quality. Third, due to the cost of training models at scale, the focus on scale produces a high barrier to entry and an environment in which research teams might be relatively risk-averse. Both of these factors can potentially lead to scientific stagnation. Again, we propose that both of these downsides can be mitigated by devoting efforts toward building and training models at smaller data scales. These models can be prototyped and tested more quickly and cheaply, allowing for broader participation and faster innovation in machine learning research.

In the rest of the paper, we present several contributions made in response to the above concerns and proposed solutions. We present a summary of and key findings from the BabyLM Challenge (Warstadt et al., 2023b), a shared task organized by the authors of this paper that challenges participants to train language models on the amount of data available to a typical human language learner. The BabyLM Challenge was held at a large Natural Language Processing conference in the fall of 2023, and received a large number of participants as well as national press coverage (Whang, 2023). We identify several key technical findings from the challenge and discuss their implications for psycholinguistics research: First, we recommend two model architectural choices as good starting points for small-scale language modeling—LTG-BERT (Samuel et al., 2023) and Contextualizer Xiao et al. (2023). Second, we identify a training approach, called curriculum learning, as generally ineffective. Note that all models were trained on English text and the extent to which these findings generalize across typologically diverse languages is therefore an open question.

The BabyLM Challenge raised several questions that cannot be answered by analyzing the performance of submitted models alone. To answer these questions, we therefore conduct a series of experiments testing hypotheses raised during the challenge. Specifically, we investigate the role of the number of training epochs (i.e., how many times a model sees its training data) for scaled-down model performance, as well as a controlled comparison between the BabyLM winning submission (ELC-BERT), and a more simple architectural variant on which it is based (LTG-BERT). For the first experiment, we find that a large number of epochs is not necessary for successful small-scale language modeling results, among the

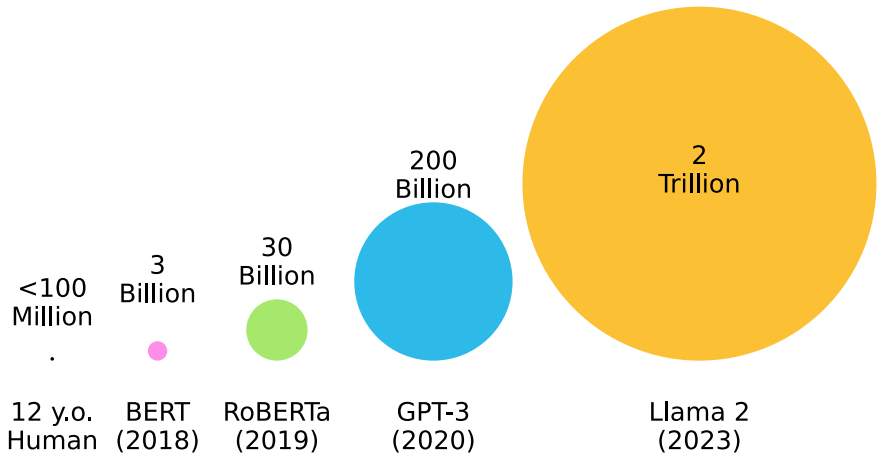


Figure 1: **Data Scale:** Modern Language Models are trained multiple orders of magnitude more word tokens than the amount available to a typical child. This image is based on Fig. 1 from Warstadt and Bowman (2022).

architectures tested. For the second experiment, we find that the simpler, baseline architecture performs as well as the BabyLM winning architecture.

2. Scaling of Neural Network Language Models

In the second half of the 20th century, natural language processing (NLP) technologies were engineered by connecting a series of highly articulated, domain-specific components. In a machine translation system, one component might be responsible for aligning words between the source and target sentence, another component responsible for homonym disambiguation, and another for scoring the naturalness of the proposed text (Block, 1962; Brown et al., 1993). But in the last decade, this paradigm has changed. Nowadays, the best-performing NLP tools typically consist solely of a language model (LM), a data structure that has been automatically learned via a training algorithm. During training, a language model must optimize its parameters to predict the probability of a unit of text (or *token*) given its preceding context.[3] In order to adapt the underlying LM to a variety of applications, after its initial training, called **pretraining**, it is trained on a secondary objective—for example, to predict the sentiment of a sentence. This

[3]Sometimes, "language model" refers to a model that has access to both the preceding and following context—most often, these are *masked language models*, like BERT. While this does not fit the classic definition of a language model, current usage includes such systems.

second training is called **fine-tuning**. Therefore, rather than a series of modular components, this paradigm involves just one single system, the LM, which can be adapted to a variety of tasks (e.g., Devlin et al. 2019). In the previous paradigm, improving the system meant improving one of its component parts, but for neural networks, breaking the model down into parts and attempting to improve each in a modular fashion is less straightforward. After all, each component, or neuron, can be connected to thousands or tens of thousands of other components at each layer of the network. So how does one increase the performance of such systems?

This is where the field of NLP has benefited from a larger trend in computer science—namely, the growing amount of data and computing power (Schaller, 1997; Coffman and Odlyzko, 2002). Neural-network-based systems that could learn on larger amounts of data tended to outperform their competitors, even without architectural changes compared to those competitors. For example, GPT-2 (Radford et al., 2019) improved substantially over its predecessor, GPT (Radford et al., 2018), even though the two have very similar architectures; this is because GPT-2 has more parameters and was trained on more data. More broadly, one reason why transformers such as GPT and GPT-2 are so successful is that they were designed to improve with increases in training dataset size. Repeated experiments all pointed towards the benefit of scaling, not only in Natural Language Processing but in other domains such as vision, leading to the so-called "bitter lesson" (Sutton, 2019): namely, that the best learning methods are general-purpose methods that can leverage the most data and compute.

But this observation—roughly, that bigger is better—raised several important questions, particularly because "bigger" is under-specified. When training deep learning architectures, three high-level elements need to be balanced: the size of the model (i.e., the number of parameters), the size of the training data, and the number of computations that are performed during training.[4] *Scaling* refers to the practice of increasing these three components in a balanced way to achieve the best model possible given one's constraints. Research in "scaling laws" (Kaplan et al., 2020) has yielded empirical discoveries for how to do this most effectively.[5] A

[4]Modern computers store real numbers in data structures call floating-point numbers. Compute costs are therefore measured in the number of floating-point operations per second, or FLOPs. In this article, however, we will discuss compute costs as the number of times a model iterates over its training dataset during training.

[5]Scaling laws are not laws in the physics sense—for one, scaling laws are different per architecture, whereas physical laws are universal. Rather, neural scaling laws refer to highly predictable reductions in loss (or, equivalently, improvements in word prediction abilities) when increasing the

growing body of work has explored this question in recent years, both for natural language technologies (Hoffmann et al., 2022; Muennighoff et al., 2024) as well as other fields (Hesslow et al., 2022; Zhai et al., 2022). Recent results suggest that, given a fixed compute budget, model architecture size and training data size should scale proportionally (Hoffmann et al., 2022). This has led to an ever-growing reliance on larger and larger training datasets, with current state-of-the-art models in 2024 trained on over a trillion words of text. The increasingly larger training data scales are visualized in Figure 1, which compares the training dataset size of today's LMs with the typical amount of human linguistic experience at the onset of adolescence—under 100 million words (Gilkerson et al., 2017).

3. The Downsides of Scaling for Psycholinguistics

How can deep learning architectures contribute to our scientific understanding of language? It is important to recognize that connectionist architectures were not originally developed as tools for processing and manipulating text data, but as models of human cognition: there is a rich tradition that uses these systems to answer scientific questions about human vision, language and other aspects of cognition (Rumelhart et al., 1986; Elman, 1990). In this section, we outline two examples of the types of contributions that neural-network-based language models can make for linguistics research (see also Linzen, 2019). We argue that these contributions are only valid under certain conditions, which are often violated by large-scale models. As stated above, we focus on the role played by language models, both because these models are particularly relevant for psycholinguistics research and because the issue of scale is particularly important in this domain. However, these arguments are not limited to language models, and can apply broadly to any neural-network based architecture being used to model human cognition.

3.1. Stimulus-Poverty Arguments

The first type of contribution uses neural networks to assess stimulus-poverty arguments. Stimulus-poverty claims are used to argue for a particular perspective on how children learn language and have been influential in the linguistics literature since they were first introduced around fifty years ago (Chomsky, 1965, 1979). Stimulus-poverty arguments point out that the primary linguistic data available to

number of parameters and training corpus size.

children are compatible with a large number of hypotheses about how that data is underlyingly structured, including many generalizations that are not observed in the "target" language the child is trying to learn, or in any natural language. However, despite this ambiguity, children routinely arrive at the correct linguistic generalizations associated with their target language. The argument goes that this successful learning cannot be driven by patterns in the data—after all, the data are ambiguous. Therefore it must be due to an innate learning preference in the child. The perspective that children are guided by inherently endowed learning constraints is known as the **nativist perspective** on language acquisition (Clark and Lappin, 2010). Stimulus-poverty arguments also point to the rapidity with which children learn language as evidence that human infants do not entertain a large number of (eventually) incorrect hypotheses about their language. This suggests, again, that children are driven by inborn learning biases.

Neural networks, and particularly language models, can inform this argument by offering one type of empirical evidence against stimulus-poverty claims (Lappin and Shieber, 2007). If it can be shown that an artificial learner can acquire the correct generalizations about a language without any linguistically-informed learning biases, then it demonstrates that, in principle, this is possible for a human language learner as well. Crucially, such evidence doesn't prove that children learn language without an innate learning bias. Rather, it shows a central claim of the stimulus-poverty argument—namely, that learning is impossible without a bias—to be false, therefore invalidating the argument. More broadly, neural network language modeling provides a lower bound on what is learnable from data by a domain-general, flexible learner. For a deeper discussion of the role of neural network modeling in stimulus-poverty claims, please see the discussions in Wilcox et al. (2023a); Warstadt et al. (2020b); McCoy et al. (2018); Yedetore et al. (2023).

How does scaling impact a model's ability to contribute to stimulus-poverty arguments? As argued in Warstadt et al. (2020b), neural networks can only disprove stimulus-poverty claims if they are no more advantaged than a human learner, with respect to both their inductive biases and their training data. If the network has super-human resources, then successful learning of a particular linguistic phenomenon no longer implies that this phenomenon is learnable, in principle, by a human language learner. After all, the network may be relying on its super-human capabilities in this case. In practice, how humanlike does a model learner have to be for its behavior to bear on stimulus-poverty claims? This is an active area of debate. There are some features inherent in neural network modeling that are unhumanlike but necessary given the current computational paradigm. For example, people learn language in social, interactive environments, but current effective techniques

for language model pretraining involve training the LM to predict a word from its context, without feedback from interactions with other agents. However, one key area that *can* be controlled and is roughly comparable to a key feature of the human linguistic environment is the amount of linguistic experience the model receives during training.

Human language learners are exposed to approximately 3 to 7 million words per year (Hart and Risley, 1995; Gilkerson et al., 2017). Therefore, by the time a child turns 12, at which they have a decent level of linguistic competence, they have experienced up to 100 million words. In comparison, many of today's language models have been trained on multiple orders of magnitude more data, with some models seeing multiple trillions of words over the course of their training. It is fair to say that this massive gap in data scale counts as one such superhuman advantage that we would like to avoid. The difference in data scale is critical, specifically for stimulus-poverty arguments, because they are based on the premise that certain constructions are so rare that a child might never encounter them over the course of language learning. It is not sound argumentation to disprove such an argument using a language model that is trained on thousands of times more data than any person will experience throughout their entire lifetime.

An additional reason why scaled-up language models bear less on stimulus-poverty arguments has to do with training data genre and quality. Large language models are trained on datasets of text scraped from the internet, whose content is often either propriety or else poorly cataloged. Linguistics and cognitive science textbooks or articles that discuss issues of learnability and give key examples may be included in these training datasets. Therefore, while models trained at smaller data scales have played an important role in assessing stimulus-poverty claims, the continued focus on bigger and bigger models means that recent advances in language modeling bear less and less on these issues.

3.2. Testing Probability-based Theories of Language Processing

The second type of contribution uses language models to empirically test theories of language processing that rely on probability distributions over words. In particular, language models have been important for developing and refining theories for the role of probabilistic prediction in language processing. As an example, we will discuss the impact that language models have made on the development of **surprisal theory** (Hale, 2001; Levy, 2008). Since scientists first started recording language processing behaviors, it has been widely observed that words which are less predictable in context are more difficult to process (Ehrlich and Rayner, 1981; Staub, 2015). Surprisal theory formalizes this observation by

hypothesizing that the effort it takes to process a word is a (linear) function of its information content, or **surprisal**. (The surprisal of a word is its in-context negative log probability, i.e., $s(w_t) = -log_2\, p(w_i \mid w_{1\ldots i-1}$.) Previously, surprisal theory was tested using non-neural-network based n-gram models (Smith and Levy, 2013) (although note that Hale (2001), which originally proposed surprisal theory, used a probabilistic context-free grammar (PCFG) language model). While such studies provided important early validation of the theory, those that used n-gram language models suffered from several setbacks, the most important being that the models used to estimate probabilities had a fixed window length, meaning that any word more than 5 words back was not factored into the estimate.

The advent of neural-network-based language models enabled researchers to collect more accurate probability estimates, enabling a more rigorous empirical assessment of surprisal theory. As a result, the relationship between word-level probabilities and human language processing behaviors has seen a surge of interest in the last five years: Using estimates from language models, studies have validated the linear relationship between word-level surprisal and reading time (Shain et al., 2022; Wilcox et al., 2023b), while others have challenged this original finding (Hoover et al., 2022; Meister et al., 2021; Brothers and Kuperberg, 2021). Other studies have investigated the surprisal–reading time relationship for cases where people read grammatically incorrect or implausible material, finding that reading times and surprisal values are poorly matched in these cases (Van Schijndel and Linzen, 2021; Wilcox et al., 2021; Arehalli et al., 2022; Huang et al., 2024). Recent work has gone beyond word-by-word reading times, and used estimates from neural network models to argue that probability-based measures underlie decisions to skip words during reading (Pimentel et al., 2023) or regress to a previous word (Wilcox et al., 2024). Looking beyond linguistic processing, studies have used neural-network-based architectures to investigate the relationship between statistical co-occurrence and syntactic structure (Futrell et al., 2019; Hoover et al., 2021). The common theme between all these works is that each uses neural network based language models to estimate underlying word-level probability distributions, which can then be used to better empirically test theories of language processing.

How does the bigger and bigger trend of language modeling put this type of contribution in jeopardy? As language models grow in terms of architecture size and training data, the way that they store information is increasingly different from those of people. To illustrate this point, it has been shown that language models memorize large passages of text from their training data and will often repeat this text verbatim during generation tasks (Carlini et al., 2023), something that people do not do during natural language production (although they are certainly capable

of such tasks, e.g., actors memorizing a script). This tendency towards long-form memorization as well as some similar types of biases, such as memorizing details only in certain contexts (Yehudai et al., 2024), suggests that, while better at language modeling, bigger models are worse for providing humanlike probability distributions that can be used to further psycholinguistic theories.

A recent line of work has clearly demonstrated the disadvantage of bigger models when it comes to modeling incremental reading times. To do so, Oh and Schuler (2023) and Shain et al. (2022) measure a model's predictive power—in other words, how well surprisal values predict reading times. They show that as models improve (i.e., their perplexity decreases and their ability to predict the next word improves), their predictive power increases, which had been observed previously (Goodkind and Bicknell, 2018; Wilcox et al., 2020), albeit not for all languages (Kuribayashi et al., 2021). However, at a certain point, the trend reverses. Oh and Schuler and Shain et al. observed that many of the models released in the past few years, which achieve state-of-the-art performance on a variety of natural language processing tasks, are actually worse than their smaller-scale counterparts at predicting human reading times. The explanation is likely that these models are very good at predicting low-frequency words, thus predicting faster reading times for these items than is observed in the human data (Oh et al., 2024). While it is not necessarily the case that smaller-scale models produce more humanlike distributions, this work suggests that, in practice, smaller models are optimal for the types of studies described above.

4. The Downsides of Scaling for Natural Language Processing

Effective NLP technologies have the potential to benefit society in several ways: They can automate expensive and time-consuming language-related tasks such as translation, summarization, document review, and copy editing, to name a few. In addition, they can serve as natural-language interfaces for technological systems, for example turning natural language queries into structured database searches, or turning natural language directions into a route plan for an automated vehicle. If successfully implemented, these capabilities will allow a wider variety of users to access technological systems, therefore broadening the positive impact these systems can have across society. Of course, the current focus on large-scale language modeling is popular for a reason: it is highly effective. Language-related tools have improved dramatically over the past half-decade, and this has been, in large part, due to the effective scaling of their underlying neural network

based architectures and training datasets. Nonetheless, scaling is not without its downsides; we outline three here.

Lack of Data-Efficient Models. Extensive research has investigated how to make language model fine-tuning (Houlsby et al., 2019; Hu et al., 2022; Dettmers et al., 2023) and inference (Zafrir et al., 2019; Dettmers et al., 2022; Hoefler et al., 2021) more parameter-efficient. Here, running **inference** on a model means obtaining its predictions, which can be used to produce generated text, for example, in chatbot applications. By **parameter efficiency**, we mean efficiency in terms of the size of the model. Because inference costs scale with the number of users in commercial settings, there has been a greater emphasis on reducing inference costs than training costs. The result of this economic incentive structure is less focus on data-efficient language modeling. However, data efficiency is becoming increasingly important as the world is running out of new high-quality text data on which to train systems. While more data is being created constantly, scaling research has found that one needs an order-of-magnitude more data (and parameters) to produce a linear increase in a model's capabilities on NLP tasks (Kaplan et al., 2020). Therefore, the pace of performance gains is likely to significantly decrease in the coming years, unless more data-efficient methods can be developed, or become prohibitively expensive to maintain.

Opacity and Controllability. Current at-scale datasets are opaque and expensive to control. An opaque dataset is one whose properties and composition are not well understood. Although there have been recent calls for better dataset documentation (Gebru et al., 2021), most state-of-the-art LLMs are trained on datasets that are proprietary, and therefore fully opaque, or whose properties are understood at only a very high level. For example, the creators of The Pile (Gao et al., 2020), a large and (admirably) open-source pretraining corpus, report that it is about 97% English, but say they cannot provide a reliable estimate of which other languages are represented in the dataset. Controllability refers to how easy it is to manipulate the contents of a dataset, for example, to remove toxic or harmful words, or to perform controlled interventions in order to run experiments. Because current datasets are so large, it is both expensive and time-consuming to modify them, and because they are so opaque, it is not guaranteed that any given intervention will successfully change all of its intended targets. Because of this, much of the research on removing toxicity and bias in language models has focused on methods to change the trained model, rather than on changing the dataset on which it is trained (e.g., Wang et al., 2022; Leong et al., 2023; Ouyang et al., 2022).

Barrier to Entry and Homogeneity. Scaling produces a high barrier to entry for what is considered cutting-edge research. The training budgets for current large-scale language modeling projects run into the hundreds of millions of dollars, due to the required personnel, computer hardware, and energy costs (Sevilla et al., 2022; Strubell et al., 2019). This results in a homogeneity of research directions, as those who can afford to participate in this research tend to be large-scale private industry groups funded by large technology corporations. Additionally, this high-risk research program can produce a risk-averse research culture: if it costs significant amounts of money and compute to produce large language models, groups will be more likely to focus only on methods that are highly likely to succeed. This increases the likelihood of scientific stagnation.

There are several proposed ways to broaden and democratize language model pretraining, including distributed (Dean et al., 2012) and federated (McMahan et al., 2017) training methods. However, one additional benefit of scaled-down pretraining over these other training approaches is that, rather than splitting the training cost between multiple parties, scaled-down pretraining simply *lowers the cost altogether*. This means that new architectures and methods can be prototyped and tested quickly and cheaply, and that coordination between groups is not necessary. Thus, scaled-down pretraining is a beneficial paradigm alongside distributed and federated training as a way to democratize and broaden participation in NLP and ML research.

5. Our Recommendations for Human-scale Language Modeling

Scientific progress that takes advantage of the synergies between psycholinguistics and NLP will require a dedicated focus on data-efficient and human-scale language modeling. Below, we outline several concrete proposals for how this can be accomplished:

- **A curated set of cognitively-inspired training datasets.** Datasets should be at "human scale," i.e., commensurate with the amount of experience people receive within a given domain Linzen (2020). Datasets should focus on the types of language people experience in their day-to-day lives—i.e., not just text, but also audio, transcriptions of audio, and/or aligned multi-modal data. Two crucial types of data that are currently under-served are aligned text–image and text–video datasets. These datasets should be open-source and well-documented—for example, with datasheets (Gebru et al., 2021).

- **A curated set of "preferred models" for psycholinguistics research.** The models in this set should be open-source, easily accessible, and available in multiple languages. They should be trained on open-source datasets whose properties are well known, such as the cognitively-inspired ones described above. Scripts should be available to easily extract word-level probabilities from these models, enabling broad access in the psycholinguistics and linguistics communities.

- **Incentives for data-efficient and small-scale language modeling research.** Incentive structures should be developed to encourage research that explores data-efficient pretraining. Incentives could include workshops or shared tasks, such as the BabyLM Challenge discussed below, but also special issues of journals dedicated to human-scale pretraining (such as the issue in which this article is published!)

In the next sections, we report on two efforts undertaken by the authors to implement the above recommendations.

6. Incentivizing Human-scale Language Modeling: The BabyLM Challenge

Data Availability. For 2023 BabyLM Challenge resources, please see the repository for last year's evaluation pipeline, which contains all code and data used in evaluating submissions. All results are hosted on DynaBench at this URL. The BabyLM training corpora are available here.

The BabyLM Challenge was a shared task that asked members of the NLP and psycholinguistics community to train a language model on the amount of linguistic data available to a human language learner, roughly 100M words or less. The challenge was held in December 2023 at CoNLL (the SIGNLL Conference on Computational Natural Language Learning). In this section, we provide an overview of the challenge, describe the various approaches taken by participants, and highlight key findings about effective strategies for pretraining language models on a human-sized dataset. For more details on the challenge, please refer to the original call for papers (Warstadt et al., 2023a). For more details on the submissions, please refer to the findings paper (Warstadt et al., 2023b).

6.1. The Structure of the Challenge

Tracks. Submissions to BabyLM were required to conform to one of three sets of guidelines, termed **tracks**. The three tracks were ***Strict***, ***Strict-Small***, and ***Loose***.

Dataset	Domain	# Words		Proportion
		Strict-Small	*Strict*	
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., 2016)	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 (Lison and Tiedemann, 2016)	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 1: The datasets we released for the *Strict* and *Strict-Small* tracks of the BabyLM Challenge. We present the number of words sampled from each of the sub-corpora that we include.

[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/

Participants in all tracks were allowed a constant number of English-language training tokens—100 million in *Strict* and *Loose* and 10 million in *Strict-Small*—to be used in total for all software used in the pipeline. This data was released by the organizing committee and is described, in detail, in Section 6.2. *Loose* track submissions were encouraged to train on data beyond the linguistic text data provided through the shared task, for example by conducting additional training on speech audio signal, code, music, or visual input. The *Loose* track also permitted the use of expert-annotated data, but any language data used to train the language model or auxiliary models counted towards the 100M word budget. For example, a *Loose* track submission could train a parser on the Penn Treebank (Marcus et al., 1993) which could be used to parse the pretraining corpus, as long as the number of words in the Penn Treebank plus the words from the pretraining corpus used by the submission totaled less than 100M.

Language model training can involve making several passes over its dataset, where each pass is called an **epoch**. For the challenge, participants were allowed to train for as many epochs as they wished. That is, multiple passes were not counted towards the 100M or 10M budget. While it is unlikely that humans process data iteratively in a manner similar to epoch-based training, there is evidence that humans do repeat some of the information they process to themselves, for example in memory replay (Carr et al., 2011). This being said, the impact of epoch count was one important question raised by the challenge, which we address in experiments reported in this paper, in Section 7.

6.2. *Creating a More Cognitively-Plausible Pretraining Corpus*

Language model training datasets are typically constructed from internet scrapes. They include text downloaded from web pages, online resource sites such as Wikipedia, and forums such as Reddit. In addition, they may include a large amount of non-linguistic content, such as computer code. As part of this challenge, we created a pretraining corpus that deviated from this typical composition and was inspired by the input to children during language acquisition, which we refer to as the BabyLM Corpus. The contents of the corpus are summarized in Table 1. For more detailed descriptions of the respective data sources, please see Appendix A of Warstadt et al. (2023b). Submissions to the *Strict* track were required to train exclusively on this corpus. Submissions to the *Strict-Small* track were required to use a scaled-down version of the dataset, approximately 10% the size of the *Strict*-track corpus, with data sources kept in the same proportions as the full 100M word corpus. Our goal was not to create a dataset that was fully faithful to the developmental experience—which includes complex social interaction, as well as huge amounts of visual information—but rather to push current pretraining datasets in the direction of cognitive plausibility. When assembling the data, we considered a variety of factors:

Dataset size. The pretraining corpus for the *Strict* track contained under 100M words, and the corpus for the *Strict-Small* track contained under 10M words. Children are exposed to at most 10 million words a year (Hart and Risley, 1995; Gilkerson et al., 2017). Choosing the beginning of adolescence (age 12) as a cutoff, therefore, the dataset should be around 100M words. The 10M word *Strict-Small* dataset corresponded to the amount of input in the first two-to-three years of development.

Text Domain. The majority ($\approx 56\%$) of the pretraining corpus was sourced from transcribed or scripted speech. This choice was made because much of the input to the typical child comes from face-to-face interaction, either through speech or sign (though this proportion decreases with age as consumption of written media increases). The speech/sign–first experience of human language learners contrasts with standard LM training corpora, which consist mostly of text that was intended to be read and is likely edited in many cases. The choice to use transcribed speech may be particularly relevant when it comes to grammar learning, as some grammatical constructions, such as nominalizations and passives, are far more frequent in writing, while others, such as first- and second-person pronouns, are more frequent in speech (Biber, 1991).

One other consideration was the genre of the transcribed speech. Child-directed speech has been used as the sole or primary data source in some previous work aiming to model child language acquisition with LMs (Reali and Christiansen, 2005; Perfors et al., 2011; Pannitto and Herbelot, 2020; Huebner et al., 2021; Yedetore et al., 2023). While it is not necessarily true that all children have access to a large amount of child-directed inputs (as separate from overheard adult-to-adult interactions), many researchers hypothesize that children will learn particular words or structures more quickly given access to simpler child-directed inputs (see, e.g., Foushee et al. 2016, Shneidman and Goldin-Meadow 2012). That said, children are routinely exposed to adult-to-adult interactions, and the extent to which adults vary their language when speaking to children varies greatly between cultures and socio-economic groups (Cristia et al., 2019). Accounting for these considerations and the availability of high-quality child-directed speech/text, about 40% of the data in the BabyLM Corpus came from sources either intended for children or appropriate for children, including child-directed speech, children's books, educational videos, and simplified English. Note, however, that we still do include sources written for adult readers, including Wikipedia articles and selections of books from Project Gutnberg.

6.2.1. Preprocessing

We performed minimal preprocessing, mostly to remove document meta-data, like `XML` tags or speaker and dialog act annotations. We preserved newlines in the original texts, which sometimes used newlines to delimit paragraphs, sometimes sentences and sometimes different documents. This means that the use of newlines as separators varies across the BabyLM corpus. For sources that we did not use in their entirety, we downsampled by randomly selecting chunks of 2000 lines or longer.[6] More details about the preprocessing steps are available in Warstadt et al. (2023b). The code and instructions for downloading and preprocessing the raw data are publicly available.[7]

6.3. Evaluating Language Models' Linguistic Abilities

To evaluate the trained language models' linguistic abilities, we provided a pipeline that would automatically evaluate LMs on a wide range of linguistic tasks. The pipeline was released as a public repository on GitHub,[8] and supported

[6] We used large chunks to preserve long-distance linguistic and narrative dependencies.

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

[8] `https://github.com/babylm/evaluation-pipeline`

models that were implemented using the HuggingFace library, which is a popular library for training language models and running inference. To submit to the challenge, users were required to (i) upload a link to their model (on any file-hosting service), and (ii) provide model predictions for all test samples in a given task; we provided a template specifying the format of the predictions file. Submissions were made via Dynabench (Kiela et al., 2021), which is a open-source platform that hosted a leaderboard, ranking the submitted models by their overall score for the competition.

Our pipeline mostly consisted of well-known NLP evaluation benchmarks. Because of this, many tasks contained vocabulary that was not contained in the BabyLM corpus. To address this mismatch, we filtered each task according to its lexical content: if an example contained any words that appear less than twice in the *Strict-Small* training corpus, we excluded the example. Otherwise, each task was presented in its original format. For details on the filtered datasets, see Appendix B of Warstadt et al. (2023b).

6.3.1. Main Evaluation Tasks

Our evaluation tasks came in two different paradigms. The first—called zero-shot evaluation—relied on obtaining outputs from the pretrained models without giving them any additional instructions or fine-tuning examples. In our case, all of our zero-shot evaluations came from the BLiMP benchmark (Warstadt et al., 2020a). BLiMP consists of tasks that evaluate if the language models' predictions are consistent with the syntactic structure of English. Each example consists of a minimal pair of sentences, where one sentence is acceptable and the other is unacceptable, differing as minimally as possible from the acceptable sentence. A model is correct on a given example if it assigns a higher probability to the correct sentence in the minimal pair. We also released a supplement to the BLiMP tasks, which tests phenomena not captured by BLiMP (see §6.3.2).

In the second type of evaluation, we fine-tuned the model on a specific NLP task, such as predicting entailment relationships between sentences, by continuing to train it on several further examples. This type of evaluation is useful because during fine-tuning one can change the training objective of the model, meaning it can be adapted into a tool for assigning categories to an input or giving binary judgments. Our fine-tuning evaluations included a subset of (Super)GLUE (Wang et al., 2018, 2019), which consists of a variety of language-related NLP tasks. The majority of these tasks involve fine-tuning the model to perform classification, given an input, which consists of either one or two sentences. For example, in natural language inference (NLI), a model is given a *premise* sentence and a *hy-*

pothesis sentence and has to categorize relationship between them as `entailment`, `contradiction` or `neutral`. An example premise is *Three tall boys are playing soccer*, and a hypothesis is *Some boys play sports*. Other NLP tasks for which we fine-tuned models included sentiment classification (SST-2), where the model has to classify a text as either positive, neutral or negative; question answering (BoolQ, MultiRC); acceptability judgments (CoLA); and commonsense reasoning (WSC).

6.3.2. Hidden Tasks

In addition to the GLUE and BLiMP-based tasks, we released three "hidden" evaluation tasks a few weeks before the challenge closed. These were: a supplement to BLiMP, the Mixed Signals Generalization Set (MSGS), and an age-of-acquisition (AoA) prediction task. MSGS and the BLiMP supplement were mandatory, while AoA prediction was provided as an optional additional analysis for participants. The motivation for using these hidden tasks was to prevent our evaluations from rewarding models that learned the patterns in their training data that allowed them to perform well on BLiMP and GLUE evaluations, but could not generalize to unseen test cases. Below, we briefly describe these tasks.

BLiMP Supplement. This task included five test suites consisting of BLiMP-style minimal pairs that cover areas of linguistic knowledge not tested by BLiMP: hypernym reasoning, question formation, turn-taking, and question-answer congruence. The test suites were semi-automatically generated using manually filled templates. As with BLiMP, models were evaluated in a zero-shot manner, by comparing the probabilities of the sequences in a minimal pair, under the assumption that the acceptable sequence should be more probable than its unacceptable counterpart. For more details, see Section 5.1.1 and Appendix C in Warstadt et al. (2023b).

Mixed Signals Generalization Set. The Mixed Signals Generalization Set (MSGS; Warstadt et al., 2020b) is a text classification task that evaluates the inductive biases of language models. For a MSGS subtask, models were fine-tuned on an ambiguous training set where the labels were consistent with both a syntactic generalization and a surface generalization, and then evaluated on examples that disambiguate which generalization the model converged on (if any). Example surface features include things like sentence length, orthography, or whether or not the sentence contains the word *the*. Example linguistic features include whether or not the sentence contains an irregular past-tense form, or whether it contains a control construction. Ideally, models would be more sensitive to linguistic features than surface features, as a systematic preference for abstract linguistic properties allows models to generalize more robustly to unseen structures. The metric for

MSGS is the Matthews correlation coefficient between the model's predictions and the labels according to the linguistic generalization on the test set. A coefficient of 1 corresponds to a systematic linguistic generalization, and -1 to a systematic surface generalization.

Age-of-acquisition Prediction. Optionally, participants could evaluate on the age of acquisition (AoA) prediction task of Portelance et al. (2023). During language acquisition, children tend to acquire words at different ages. For example, *mommy* is almost always acquired before *drawer* or *green*. The question of what predictions age of acquisition (AoA) for words is a subject of ongoing research. For this task, AoA is defined as the time at which 50% of children are reported by their parents to produce a given word, using the parental reports of Goodman et al. (2008). The AoA prediction task compares LMs' word surprisals with children's AoA of the same words. Language models' surprisals are converted into an AoA score by asking to what extent they increase the predictive power of linear regression models trained to predict age of acquisition over baseline models that only include word frequency and concreteness ratings.[9] While we did not require participants to submit these scores as part of their predictions, we provided code so that they could include this score as an additional analysis point in their paper submissions. Seven teams (22.6%) evaluated on the AoA prediction task. For more results and discussion, see Appendix E of Warstadt et al. (2023b).

6.3.3. Task Aggregation

To compute the aggregate score across tasks, we weighted BLiMP and the BLiMP-supplement together at 50% (weighting all sub-tasks equally), (Super)GLUE at 30%, and MSGS at 20%. This weighting scheme was arrived at heuristically, though we did observe that the winners for each track were stable across a wide range of reasonable weightings. Our online submission portal, Dynabench, allowed users to specify a custom task weighting to compute an alternative aggregate score.

6.4. Baselines and Skylines

Baselines. To provide simple baselines for our evaluation tasks, we trained multiple models on the data released for *Strict-Small* and *Strict* tracks and evaluated

[9] It is not clear whether optimizing LM performance on this task necessarily leads to language models that can more accurately predict a word given its context. Therefore, this task was included more as a measure of how well LMs align with humans—and thus, as a measure of their usefulness as cognitive models of language acquisition—rather than as a measure of quality or performance.

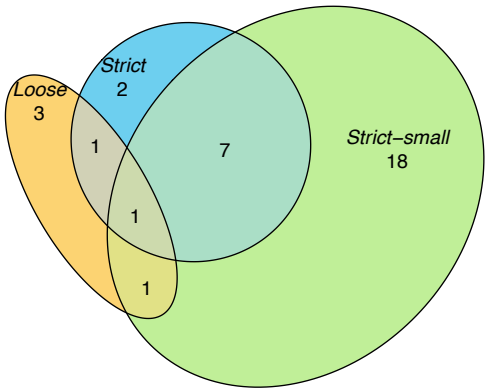


Figure 2: Number of participants who submitted to each track, with multiple submissions counted once.

them on the evaluation tasks. We provided three baselines, using popular language modeling architectures: OPT-125M (Zhang et al., 2022), RoBERTa-base (Liu et al., 2019), and T5-base (Raffel et al., 2020). For details on the architectural choices and hyperparameters, see Section 5.4 of Warstadt et al. (2023b). Although most of these hyperparameter choices were loosely inspired by Huebner et al. (2021), we expected that the specific choices could be further improved and left these potential improvements as possible topics for submissions. We found that our baseline models achieved reasonable performance on the evaluation tasks, with clear improvement from the *Strict-Small* to *Strict* datasets, and a notable gap between their performance and the performance of our skyline model, described below.

Skylines. To get an approximation of how well a larger model could perform in our task and setting, we chose two large-scale models and rand them through our evaluation pipeline. The two models we used were Llama 2 (Touvron et al., 2023) (the variant with 70 billion parameters) and the RoBERTa-base model. This was meant to provide a comparison between our BabyLM models and the state of the art in 2023. [10]

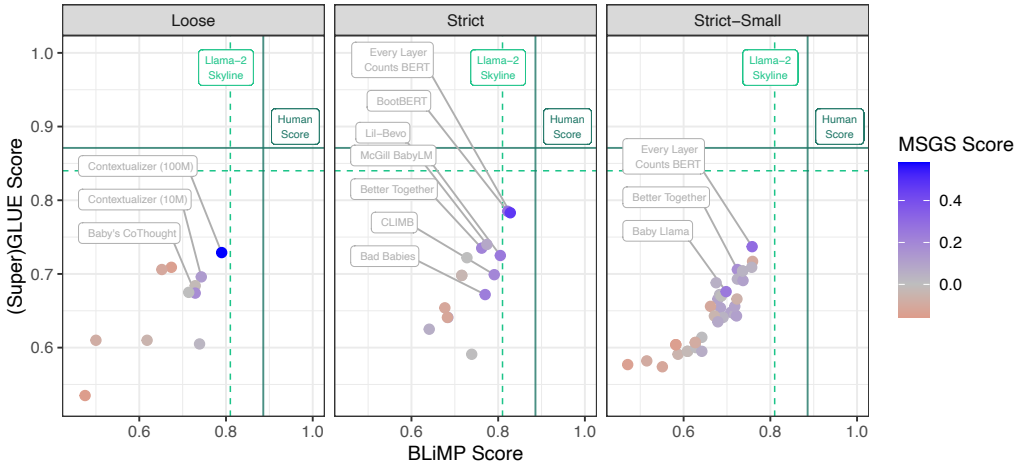


Figure 3: **Summary of BabyLM Submission Results:** Each point represents an official model submission. Scores are broken down into performance on BLiMP (x-axis), GLUE (y-axis) and MSGS (color). Submissions that achieve an aggregate score above 0.6 are labeled in gray. Green dashed lines show Llama 2 skyline performance, and green solid lines show the human performance ceiling.

6.5. Results and Analysis

We received 31 papers and 162 models in total. Some participants submitted to multiple tracks; we show data for unique participants in Figure 2. The results that achieved an aggregate score across both BLiMP and GLUE of above 0.6 are shown in Figure 3, with the scores of the top-performing models in each track detailed in Table 2. In the figure, dashed green lines show the performance of the Llama 2 skyline. Solid green lines show human performance on GLUE reported in Nangia and Bowman (2019), and human performance on BLiMP as reported by Warstadt et al. (2020a). For the GLUE benchmark, human scores are obtained by training naive crowd workers on each NLP task—for example, teaching them to classify entailment relations between sentences—as well as giving them 20 examples. For the BLiMP benchmark, scores are obtained by asking naive participants to choose between sentences in a forced-choice task, and calculating the proportion of times each individual chooses the grammatical variant.

[10]One point of difference between the Llama skyline and our BabyLM models was that we evaluated Llama 2 on (Super)GLUE using in-context learning. In this setting we get the model to solve each GLUE task by showing it a few examples and prompting it to complete the relevant test example. We do this because fine-tuning these at scale LLMs is computationally expensive.

	Model	BLiMP	GLUE	MSGS	BLiMP-Supp.	*Aggregated*
	Llama 2	0.84	0.84	0.26	0.75	0.71
	RoBERTa-Base	0.87	0.79	0.24	0.76	0.70
Strict	ELC-BERT Charpentier and Samuel (2023)	0.85	0.78	**0.47**	**0.77**	**0.74**
Strict	BootBERT Samuel (2023)	**0.86**	**0.79**	0.28	0.72	0.70
Strict	McGill-BERT Cheng et al. (2023)	0.84	0.72	0.25	0.71	0.67
Strict	*Best Baseline (OPT-125M)*	0.75	0.70	0.13	0.68	0.60
Strict-Small	ELC-BERT Charpentier and Samuel (2023)	**0.80**	**0.74**	**0.29**	0.67	**0.66**
Strict-Small	MLSM Berend (2023)	0.79	0.71	0.17	0.57	0.61
Strict-Small	McGill-BERT Cheng et al. (2023)	0.75	0.70	0.13	**0.68**	0.60
Strict-Small	*Best Baseline (OPT-125M)*	0.63	0.62	0.10	0.53	0.50
Loose	Contextualizer Xiao et al. (2023)	**0.86**	**0.73**	**0.58**	0.63	**0.73**
Loose	McGill-BERT Cheng et al. (2023)	0.80	0.68	-0.02	0.57	0.57
Loose	BabyStories Zhao et al. (2023)	0.78	0.61	0.03	**0.65**	0.56

Table 2: Top 3 systems for each track, as well as the baseline model with the highest aggregate score. We also show "skyline" models: RoBERTa-base and Llama 2 trained on their full pre-training corpora. Each task score is simply the mean score across each of its subtasks. The aggregate score is a weighted average of each task. We **bold** the highest-scoring system for each task within each track.

Below, in Section 6.5.1, we break down the submissions based on the type of approach they use, and discuss the effectiveness of these different approaches. Then, in Section 6.5.2, we discuss the winning models in each track, and what they can tell us about human language learning and processing.

However, before we discuss the details of any model or approach, we start by pointing out a few high-level takeaways from these results, starting with comparisons between the different tracks. The strongest results were achieved by models in the *Strict* track. Given the *Strict* track's larger training corpus relative to the *Strict-Small* corpus, it is not surprising that these models performed better. However, there are two interesting trends: First, *Strict* models did not outperform those in *Strict-Small* by a large amount, even though the size of training data was an order of magnitude larger. For example, there are only two models in the *Strict* track that achieve higher GLUE scores than the best-performing *Strict-Small* model. Second, models in the *Loose* track tended to perform worse in the aggregate than those in the *Strict-Small* track, even though they potentially had access to additional, non-linguistic, data. One conclusion we can draw from this is that learning from multiple modalities of data presents a challenge in its own right, and that current model architectures are not optimized to efficiently utilize multiple types of inputs during training.

The other important high-level takeaway is that many BabyLM models are

very close to the Llama 2 skyline, and also close to achieving human-level performance on BLiMP and GLUE (i.e., they are near the green lines in Figure 3). Strong performance could be expected in the case of (Super)GLUE, where models were fine-tuned with additional data, but we note that even for BLiMP, the top-performing model is only about 3% shy of human performance. Given that successful training on developmentally plausible corpora could have ramifications for cognitive and linguistic theories of learnability, these results point to two important takeaways: (1) Human-level results have not been achieved *yet*. However, (2) given the strong performance of the top-scoring models, human-level results appear likely to be achieved very soon, possibly within the next few years. Of course, one possible criticism of current metrics, like accuracies on BLiMP, is that they do not accurately measure linguistic competence. We are sympathetic to such concerns, but we also note that BLiMP, and other related syntactic benchmarks such as those presented in Marvin and Linzen (2018) and Gauthier et al. (2020), were specifically designed to mimic the types of tests invented by linguists and cognitive scientists to reveal syntactic competence in humans—i.e., they are all based on minimal pair sentences. Thus, while it is imperative to continue building more comprehensive and larger datasets, we believe it is fair to say that the close-to-human scores observed in the BabyLM challenge on BLiMP reflect genuine grammatical generalizations learned by the models. This evidence is supplemented by MSGS results, which were generally positive, indicating a preference for structural, as opposed to surface-level, syntactic generalizations.

6.5.1. Common Methods

To help us understand which approaches were effective, we hand-coded each submission based on the method(s) it employs. We show the breakdown of approaches in Figure 4 and we visualize the performance of different methods in Figure 5. For more information about the common approaches, as well as brief descriptions of submissions in each category, see Appendix A. For the purposes of the remaining discussion, the most important approaches that the reader should be aware of include curriculum learning, architectural modifications, and data preprocessing, which we define briefly below. In **curriculum learning**, the dataset is organized according to some metric, often a simplicity metric, with models first training on simpler examples before graduating to more difficult examples. It is motivated by the idea that successful learning depends on "starting small" (Elman, 1993)—or in other words, that presenting data in a random order is less helpful than presenting data in a meaningful order with gradually more complex concepts (Bengio et al., 2009). Curriculum learning has some parallels to human

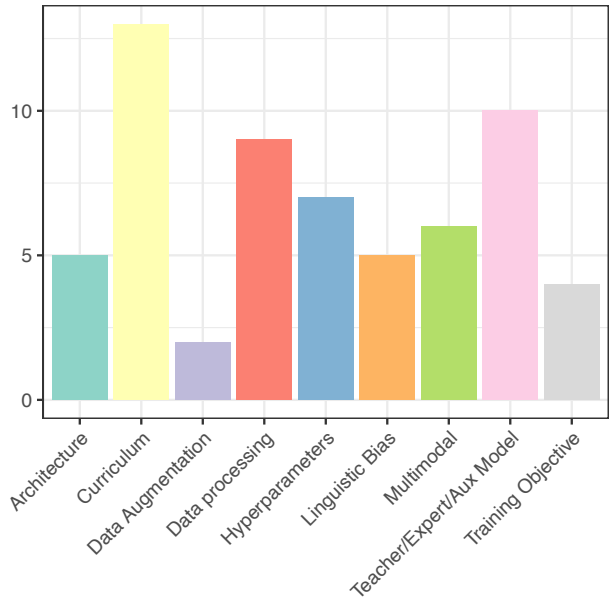


Figure 4: Total number of submitted models that used each of the nine approaches in our typology. We count at most one submitted model per participant per track.

language learning, in particular to child-directed speech, a genre of speech that is characterized by its reduced vocabulary size and simple constructions (Cameron-Faulkner et al., 2003). Some evidence suggests that child-directed speech helps language learning, especially with early vocabulary development and reading skills (Rowe, 2008). That said, other work suggests that language learning proceeds at similar paces in groups where child-directed speech is not employed as frequently (Ochs, 1982; Heath, 1983). Curriculum learning was by far our most common approach, however it was found to produce only marginal gains above baselines. **Data preprocessing** involves making modifications to the underlying data, or the way the data is presented to the model, that does not involve ranking the entire dataset according to some metric. Finally, **architectural modifications** involved producing some novel architectural innovation that went above and beyond changing existing parameters that come pre-built into current machine learning models, such as the number of training examples the model sees at any given time, or the rate at which it updates its parameters during training. Data preprocessing and architectural modifications were found to be the most effective strategies in our meta-analysis.

All of the models submitted to the competition used a pre-existing **backbone**

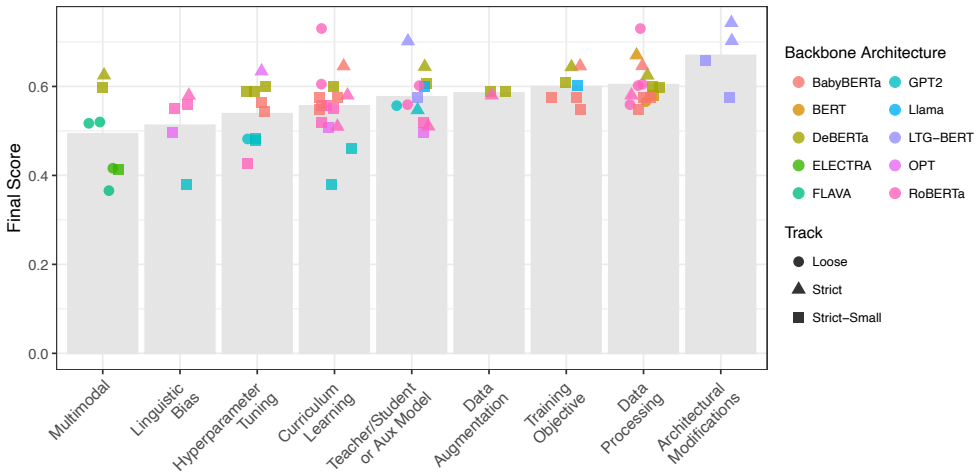


Figure 5: **Effect of Training Strategy and Backbone Architecture:** Each point represents a submission. Some submissions may appear more than once if they use multiple strategies. Shapes show the challenge track to which the model was submitted. Colors show the backbone architecture on which the model is based. Gray bars show within-category aggregates.

architecture, with many submissions based on BERT (Devlin et al., 2019), GPT (Radford et al., 2018), and Llama (Touvron et al., 2023). Although there are several differences between these models, the most significant of these is that BERT is a masked LM, meaning it predicts a word given its surrounding context, whereas Llama and GPT are autoregressive LMs, meaning that they predict a word given only its preceding context. In Figure 6, we separate models by this underlying architecture.

6.5.2. Winning Submissions

Below, we discuss the winning submissions from each track in greater detail and ask what, if anything, they can tell us about human language learning or language processing.

ELC-BERT. The winner of both the *Strict* and *Strict-Small* tracks was ELC-BERT, (Charpentier and Samuel, 2023). This model, as well as the runner-up submission Boot-BERT (Samuel, 2023), used as their starting point the LTG-BERT architecture from Samuel et al. (2023). Both of these submissions make additional architectural changes on top of the LTG-BERT backbone. Specifically, the ELC-BERT adds a mechanism, called *skip connections* or *residual connections*, for information to flow between layers of the network, allowing representations computed in early layers to directly impact computations at higher layers of the network. However, scores

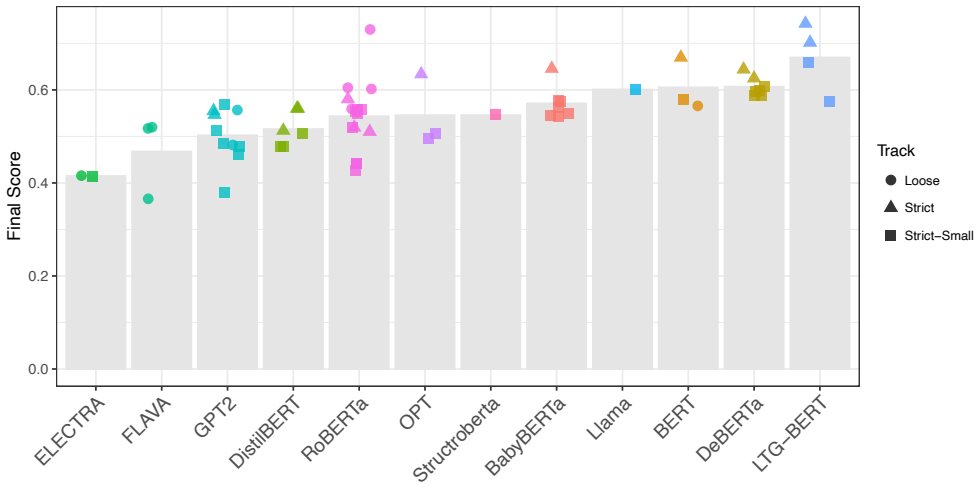


Figure 6: **Effect of Backbone Architecture:** Each point represents a submission. Shape indicates the challenge track. Gray bars show within-category aggregates.

from LTG-BERT submitted by the authors suggest that this backbone architecture plays a large role in the submissions' successes. Therefore, here, we will focus on LTG-BERT and discuss its relationship to hypothesized cognitive architectures of language learning and processing.

LTG-BERT's main contribution is a synthesis of several optimizations to the Transformer architecture, namely:

1. Layer normalization, following Shleifer et al. (2021). This procedure scales and centers the models' weights at each layer.

2. GEGLU feed-forward modules (Shazeer, 2020). Here, feed-forward layers learn two sets of weights and biases. One of these is not passed through an activation function, while the other is passed through a Gaussian Error Linear Unit (GELU) activation function, which is similar to the more standard ReLU activation function, except it is curved around zero.

3. Disentangled attention (He et al., 2021). Attention is a mechanism that allows the model to learn the relative importance of connections between its different input tokens. In most transformer models information about word content and word position are shared in a single embedding. In this attention mechanism, word content and word position are represented in separate embeddings, which are both used to compute attention weights.

4. Reducing attention layers' weight initialization following Nguyen and Salazar (2019). In this modification, weights in attention layers are initialized to be smaller than is typical. This was proposed to help model convergence, in general, not necessarily just in the small data-scale setting.

ELC-BERT includes an additional modification on top of these three. In ELC-BERT the input to each layer is a weighted sum of the outputs of all previous layers, meaning that the network can learn indirect connections between layers. Another notable property of LTG- and ELC-BERTs is that all released versions of these models have been trained for a large number of epochs. For example, Charpentier and Samuel (2023) train models for over 450 epochs for their *Strict* submission, and over 2000 epochs for their *Strict-Small* submission, which is much higher than is standard practice. By contrast, we trained our baseline models for about 20 epochs.

Taking a step back, what can these architectural modifications tell us, if anything, about human language learning? First, given how many modifications the LTG-BERT makes on top of the traditional transformer architecture, it is difficult to disentangle which changes are specifically responsible for its performance gains. Furthermore, certain aspects of the architecture are likely important because they mitigate technical issues in the procedure used to train language models, called back-propagation. For example, the layer normalization discussed above ensures that all the weights learned by the model are on the same scale, which speeds up training time and makes it more likely that the model will converge to a stable set of successful weights. These changes are unlikely to reveal anything about human language processing. However, other features of the architecture are easier to compare directly with hypothesized features of human cognitive architectures. For example, relating to the disentangled content and positional representations, it has been argued that, at least during reading, people maintain information about the locations of words on the page, separately from their content (Kennedy, 1992; Kennedy et al., 2003). This separate representation of word location has been argued to reduce cognitive load and facilitate efficient language processing, however, connections to efficient language *learning* have not been explored. Moving beyond reading, the notion of *location* doesn't have to mean location on a page. Rather, this representational approach might allow models to learn more abstract concepts about location, such as abstract syntactic roles. Decomposition along these lines—i.e., separate learning and representation of content and form—has been proposed previously as a mechanism for explaining how symbolic structures are embedded inside distributed systems (Smolensky, 1990). Therefore, while strong

performance of LTG-BERT alone tells us little about human language processing, it does point to directions that can serve as the basis for future research.

Contextualizer. The winner of the *Loose* track was the Contextualizer model of Xiao et al. (2023), which used a data processing scheme in which extra training samples are created by combining chunks of text from different sources in the dataset. Repeating this process 40 times for each chunk gives an artificially augmented dataset that has as many training samples as a four billion word dataset, but only uses 100 million words. The insight behind the Contextualizer model is that people may hear the same words, or chunks of words, over and over again, but that the varying context helps them to learn novel information about the word's, or chunk's, meaning and proper use. Furthermore, it has been argued that repetition of the same chunks in different environments helps models to learn the hierarchical structure of language. For example, if a child hears the NP *The big blue ball* as both a fragment answer to a question and also as a fronted element in a sentence, this may provide evidence that it counts as a single constituent. Therefore, this type of data augmentation has been previously argued, e.g., in Andreas (2020) to give models a bias towards compositionally by teaching them that chunks can be recombined in different ways.

McGill-BERT. This submission from Cheng et al. (2023) was runner-up in the *Strict* and *Loose* tracks. The authors improve over the original BERT model by modifying two features: First, they shorten the context window, so the model only learns more local relationships between words. Second, they modify the way that training examples are presented to the model, splitting up examples into individual sentences, rather than in chunks that may contain multiple sentences. Rather than telling us something about psycholinguistic processes, the authors propose that this regime is particularly well-suited to the BabyLM training dataset, in particular its CHILDES portion. Because we include only child-directed utterances of CHILDES, and remove any intervening child-produced utterance, each sentence does not necessarily follow from the previous one. Therefore, learning to predict these sentences separately, rather than as a single cohesive unit, may constitute an easier learning task.

CLIMB: A compelling negative result. In addition to track winners, we gave several awards to outstanding papers, one of which was "CLIMB: Curriculum Learning for Infant-inspired Model Building" (Martinez et al., 2023). This work proposed a typology for, and conducted a thorough evaluation of curriculum learning. The authors vary the curriculum based on three features: First, the authors experiment

with several vocabulary curricula, in which models begin training over a simple vocabulary that slowly grows in size. For example, an early vocabulary might only consist of frequent words or nouns. Second, the authors experiment with curricula based on data difficulty by defining ways to measure difficulty both from an objective function (e.g., how long is the sentence?) and also from the perspective of the model (e.g., which unseen example does the model find the most likely?). Third, the authors explore curricula based on a model's objective function, in particular, the level of granularity at which the model must make its prediction, for example, making predictions over words vs. parts of speech.Although Martinez et al. find that none of the curricula within this typology leads to widespread improvements across the evaluation tasks, the exhaustiveness of this search and the careful controls and baselines in the study make this negative result a valuable contribution. Namely, it suggests that curriculum learning is unlikely to be effective for developmentally plausible language models, at least in its current form.

We take these overall negative results for curriculum learning as fitting into an ongoing debate about the role of cognitive and data limitations in language learning. The main locus of this debate is the *less is more* hypothesis (Newport, 1988), which suggests that children's cognitive limitations force them to attend to smaller, compositional linguistic units, which is overall beneficial for the learning process. This hypothesis has been supported by several types of experimental studies, including studies showing that reducing adult's cognitive capacities makes them learn in a manner closer to children (Cochran et al., 1999). Evidence for the less is more hypothesis has also been found in computational modeling studies, for example, Elman (1993) found that a simple recurrent network can learn the patterns of English embedded clauses, but only if trained initially on simple sentences that did not include embedded clauses, or on networks that were initially memory constrained. This led Elman to suggest "starting small" as an approach for successful language model training. However, the model in Elman (1993) was trained on an artificial version of English. Running similar tests on more realistic datasets, Rohde and Plaut (1999) do not find evidence that "starting small" is beneficial to performance. Rather, they find that withholding complex examples at the beginning of training can hinder language learning in connectionist models. We take the negative results from Martinez et al. (2023), as well as other BabyLM submissions, as being in line with the conclusions of Rohde and Plaut (1999). They suggest that simplifying the early stages of LM training does not result in better learning outcomes, at least for small-scale datasets. We note, however, that these results should not be taken as evidence against the less is more hypothesis in humans. Just because the language models tested do not benefit from simplicity

early in training does not mean that children do not benefit from it. Rather, it may suggest that the role of resource limitations may be one key difference between LMs and human children.

6.6. Interim Discussion

The BabyLM Challenge led to a number of concrete outcomes aligned with the vision for more human-scale language modeling outlined in section 5. First, it drew attention to the challenge of data-efficient language modeling, and provided a venue for dozens of participants to share ideas and resources. It produced a more cognitively plausible pretraining corpus, which will facilitate human-scale model training going forward. Finally, the results of the challenge produced a number of lessons that will help to improve future small-data language models, including the effectiveness of the LTG-BERT architecture, and the relative ineffectiveness of curriculum learning.

One interesting outcome of the challenge is that successful submissions were not directly inspired by theories from human cognition. For example, LTG-BERT is successful because of several architectural modifications, none of which have a direct basis in the cognitive science literature. (That being said, as noted above, there are several parallels between these modifications and proposals for language learning and processing architectures in people.) Similarly, the McGill-BERT submission achieves impressive performance by changing key hyperparameters, such as context length, and how the data is presented to the model. Again, neither of these takes direct inspiration from cognitive theories. It is possible to read this trend in two ways: Pessimistically, one could conclude that theories from cognitive science have little to contribute toward effective small-scale language modeling. However, more optimistically, it's worth noting that many of the submissions opted to investigate what might be low-hanging fruit—aspects of pre-existing models that are not optimized, or model architectures that differ minimally from pre-existing ones. One reason for this might have been the relatively short timeline of the task, just about 6 months from data release to when submissions were due. It is our hope that subsequent challenges can attract submissions that take more risks by modifying neural network architectures based on theories from linguistics and cognitive science.

7. Experiments

Data Availability. Please see this repository, which contains code for training LTG-BERT and ELC-BERT models on the BabyLM training corpora.

In this section, we present experiments inspired by unanswered questions from the BabyLM Challenge. In the first experiment, we investigate the role of training time, measured in the number of epochs, on the performance of LTG-BERT. This architecture was originally trained with a significantly larger-than-standard number of epochs, which arguably makes it less cognitively plausible; is such a large number of epochs necessary? In the second experiment, we directly compare ELC-BERT, which was the official winner of the BabyLM challenge, against LTG-BERT. We ask, are the skip connections between layers introduced by ELC-BERT necessary for strong performance in small-data language modeling? We find that LTG-BERT is about as good as ELC-BERT in our controlled setting, and that, while a large number of epochs can increase model performance, returns are strongly and quickly diminishing. We conclude that LTG-BERT is sufficient for successful small-scale language modeling and that it can be well-trained in about 20 epochs.

7.1. Experiment 1: Evaluating the Role of the Number of Epochs in Training

The BabyLM challenge did not place any limits on the amount of computational resources participants could use when training their models. Because our dataset size was fixed for participants in the *Strict* and *Strict-Small* tracks, this meant that computational resources fluctuated as a function of (i) model size and (ii) training epochs, or the number of times the model sees its training data. Research in scaling has determined that training data size and model size should scale proportionally (Hoffmann et al., 2022); therefore, entrants tended not to train large models. When entrants did use more computational resources, this tended to be allocated toward an increased number of training epochs. When preparing baselines, we trained models for 20 epochs, which we chose based on prior experience. We intended this number—20 epochs—to also serve as a best first guess for our participants' training budgets, especially for those who did not have extensive prior experience training language models.

While most participants did indeed train in the general range of 20 epochs, some chose to train for much longer. In particular, the creators of ELC-BERT trained for 450 epochs in their *Strict* submission and 2,000 epochs in their *Strict-Small* submission, which is well beyond typical for language modeling research. Therefore, one big unanswered question at the end of the challenge was whether these models had achieved top scores because of their architectural innovations, or rather because they had trained for longer than other models.

To answer this question, we reproduced the LTG-BERT training pipeline using publicly available code from the authors and analyzed how the performance of

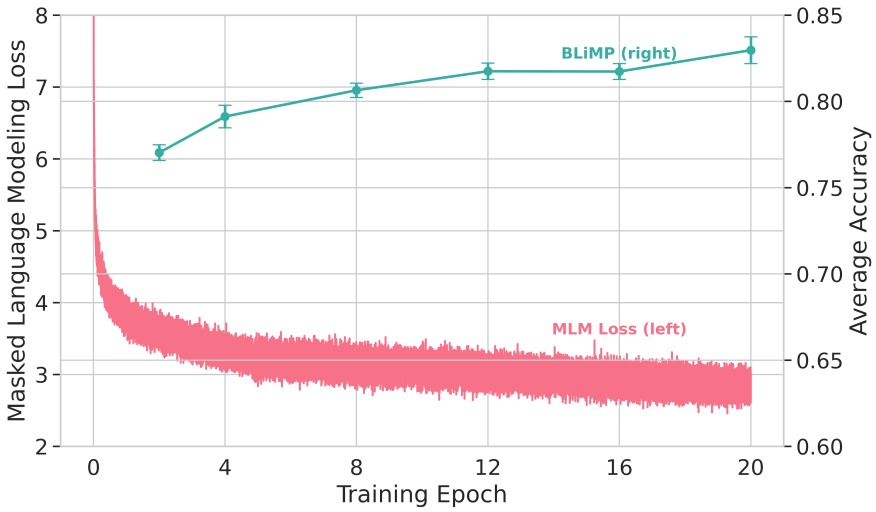
the model improved over the course of training. For the low end, we trained a model on the *Strict* dataset for 20 epochs to match our baselines. At the other extreme, we trained on the *Strict-Small* dataset for 800 epochs, to more closely match the training epochs of the original LTG-BERT based models submitted to the competition.[11]

For the training and BLiMP performance dynamics of our models, see Figure 7. We find that both the *Strict* and *Strict-Small* model's training loss decays roughly exponentially during training, typical to the training dynamics of most language models (Muennighoff et al., 2024). For both the *Strict* and *Strict-Small* models, the increase in BLiMP performance also diminishes exponentially over time. This trend also holds for the *Strict* model on GLUE (see Figure A.8), but not for the *Strict-Small* model, where GLUE performance actually decreases slightly from 50 training epochs onwards.

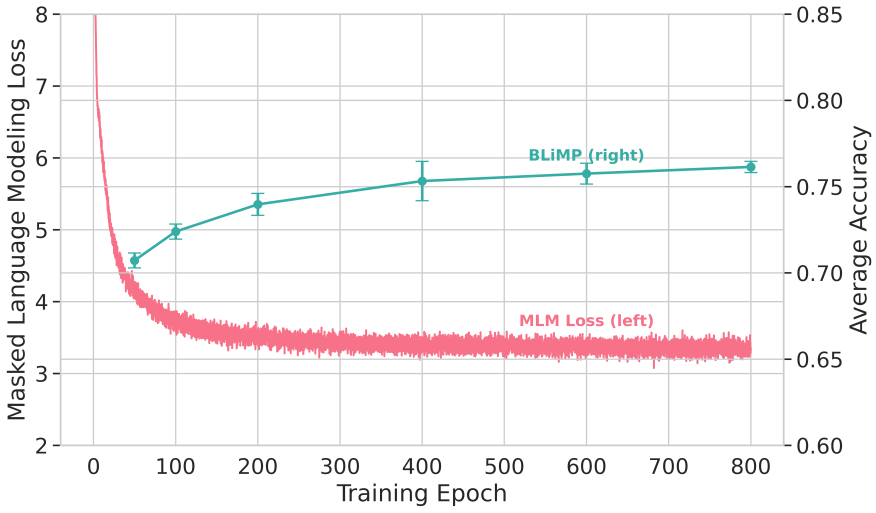
Taking our results in conjunction with that of LTG-BERT and ELC-BERT (Samuel et al., 2023; Charpentier and Samuel, 2023), we conclude that these architectures are generally robust to overfitting on *Strict* and *Strict-Small*, allowing them to be trained for large numbers of gradient updates and training epochs on small corpora. Additionally, training for longer is beneficial for downstream performance in most cases. We do not observe any unusual learning dynamics, such as sudden drops in the training loss, nor instances in which test scores improve dramatically late in training, a phenomenon observed in some small-scale synthetic data experiments (Power et al., 2022; Murty et al., 2023). Importantly, in all our experiments, both the training loss and BLiMP and GLUE performance change gradually with more epochs, with gains diminishing as training continues. This pattern of diminishing returns is in line with the previous literature on language model training (Hoffmann et al., 2022; Muennighoff et al., 2024). Despite this pattern, LTG-BERT does still perform better than our baseline models given the same number of training epochs; this confirms that the architectural innovations of LTG-BERT lead to improvements on our evaluation tasks.

We derive two conclusions from this experiment: First, LTG-BERT can be trained to achieve good performance with less computational resources and a smaller batch size than reported in the initial paper. Second, the returns of additional training time quickly diminish. Specifically, we show that the 500 training epochs

[11] Although we used fewer training epochs, our batch size was also smaller than the one reported in the original LTG-BERT paper due to computing constraints. Therefore, the number of gradient updates, or times when the model updates its weights based on the observed training data, is *actually higher than that of LTG-BERT.* See Appendix B for details.



(a) *Strict*, 20 epochs



(b) *Strict-Small*, 800 epochs

Figure 7: Training curves and BLiMP evaluation scores for *Strict* and *Strict-Small* LTG-BERT. All losses and scores are averaged over 3 random seeds. The Pearson correlations between training loss and BLiMP performance are -0.99 and -0.95 for *Strict* and *Strict-Small* respectively, indicating strong linear relationships. In other words, the training loss and BLiMP evaluations improve at roughly the same rate.

	Model	BLiMP	GLUE	MSGS	BLiMP-Supp.
	Llama 2	0.84	0.84	0.26	0.75
	RoBERTa-Base	0.87	0.79	0.24	0.76
Strict	ELC-BERT Charpentier and Samuel (2023)	0.85	**0.78**	**0.47**	**0.77**
Strict	LTG-BERT Samuel et al. (2023),	**0.86**	**0.78**	0.28	**0.77**
Strict	[R] ELC-BERT, 20 epochs	0.83	0.75	0.25	0.67
Strict	[R] LTG-BERT, 20 epochs	0.83	0.76	0.19	0.68
Strict	*Best Baseline (OPT-125M)*	0.75	0.70	0.13	0.68
S-Small	LTG-BERT Samuel et al. (2023)	**0.80**	**0.74**	**0.29**	**0.67**
S-Small	[R] LTG-BERT, 800 epochs	0.76	0.67	0.02	0.63
S-Small	*Best Baseline (OPT-125M)*	0.63	0.62	0.10	0.53

Table 3: A comparison between our reproductions of LTG-BERT and ELC-BERT (labeled "[R]"), our baselines, and existing results.

of the original paper are not necessary and that good results can be achieved with about 20 epochs of training.

7.2. Experiment 2: Comparing LTG-BERT and ELC-BERT

Another unanswered question from the challenge relates to the relative importance of the LTG-BERT baseline versus the skip connections introduced for ELC-BERT (described in §6.5.2). To address this question, we train ELC-BERT for 20 epochs on the *Strict* dataset, and compare its performance to that of LTG-BERT. Due to the significant cost of evaluating intermediate checkpoints, we only examine the final trained model for ELC-BERT. The final results of this comparison, as well as the final scores for our LTG-BERT models trained for the previous experiment, can be seen in Table 3. We find that the performance of the two models is similar. ELC-BERT achieves higher scores on MSGS, but LTG-BERT is better for GLUE and the BLiMP supplement. We take these results to suggest that the LTG-BERT architecture is what drives superior performance on BabyLM evaluations, as opposed to the skip connections that are added atop the LTG-BERT architecture to create ELC-BERT.

8. General Discussion

Cognitive modeling with neural networks has played an important role in psycholinguistics and in many areas of cognitive science. As neural network approaches get more and more powerful, neural network modeling stands to produce many more insights in the decades ahead. At the same time, it is important to take stock and to ask how trends shaping the development of these models

will impact their ability to help us answer scientific questions about the human mind. This paper has attempted to do just that. We have argued that, while beneficial for producing more powerful models, the current trend of scaling up has a number of potential downsides for psycholinguistics research. We recommend that linguists, cognitive scientists, and computer scientists work together to produce shared resources that are more "human-scale," including human-scale pretraining datasets and models, as well as venues that support research dissemination in this area. In addition to potential scientific impact of small-scale language modeling, we believe that focusing on such models has the potential to lower the barrier of entry for participation in language model pretraining research, allowing for a wider and more diverse set of interested scientists to contribute.

We reported on two efforts undertaken by the authors to actualize these recommendations: the BabyLM Challenge and two experiments that followed up on questions raised by the winning submissions. The most significant finding from the challenge itself is that, even at smaller data scales, current neural network architectures are very close to achieving human-level performance on many linguistic tasks. The best performing models from the challenge showed sensitivity to syntactic constraints that was on par with models several orders of magnitude their size, and were just 3% shy of human-level performance on this task. This is a significant achievement. Given the rate at which language modeling performance has improved recently, it is likely that computational models—even ones trained on human-scale datasets—will show sensitivities to syntactic constraints that are on-par with humans. Furthermore, the number of participants who contributed to the first iteration of this shared task demonstrates the broad interest in this topic. Finally, the challenge produced several concrete outcomes, including (i) the BabyLM Corpus, (ii) a series of small-scale models, and (iii) several lessons for best practices in small-scale language modeling. These include the effectiveness of the LTG-BERT and Contextualizer methods and the relative ineffectiveness of curriculum learning.

While the efforts reported here present a step toward more plausible cognitive models, there are still significant differences between today's neural network models and an ideal cognitive model that helps answer scientific questions about linguistic cognition. We argue that, currently, the most significant of these has to do with input modality. In this paper, we have generally equated human-scale with small-scale; with respect to number of words, this is fair. However, many people also receive vast amounts of *non-linguistic* visual input over the course of development. It is an open question, however, just how much visual input is useful or necessary for successful language learning. Visual information may be hard

to align with words, and people who lack vision learn language without visual input altogether. This said, creating more multi-modal text–image and text–video datasets, and designing architectures that can effectively learn from these data, are logical next steps in the creation of cognitively plausible computational models that can help us answer questions about human language learning.

Acknowledgments

E.G.W. was supported by an ETH Postdoctoral Fellowship. M.Y.H. was supported by an NSF Graduate Research Fellowship and NSF Award 1922658. A.M. was supported by a postdoctoral fellowship awarded by the Zuckerman STEM Leadership Program. A.W. was supported by an ETH postdoctoral fellowship. This work was supported in part through the New York University IT High Performance Computing resources, services, and staff expertise.

References

Abdelali, A., Guzman, F., Sajjad, H., and Vogel, S. (2014). The AMARA corpus: Building parallel language resources for the educational domain. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 1856–1862, Reykjavik, Iceland. European Language Resources Association (ELRA).

Amariucai, T. and Warstadt, A. S. (2023). Acquiring linguistic knowledge from multimodal input. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Andreas, J. (2020). Good-enough compositional data augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566, Online. Association for Computational Linguistics.

Arehalli, S., Dillon, B. W., and Linzen, T. (2022). Syntactic surprisal from neural models predicts, but underestimates, human processing difficulty from syntactic ambiguities. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 301–313.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (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.

Berend, G. (2023). Better together: Jointly using masked latent semantic modeling and masked language modeling for sample efficient pre-training. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Bhardwaj, K., Shah, R. S., and Varma, S. (2023). Pre-training LLMs using human-like development data corpus. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Biber, D. (1991). *Variation across Speech and Writing*. Cambridge University Press.

Block, H.-D. (1962). The perceptron: A model for brain functioning. i. *Reviews of Modern Physics*, 34(1):123.

Borazjanizadeh, N. (2023). Optimizing GPT-2 pretraining on BabyLM corpus with difficulty-based sentence reordering. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Brothers, T. and Kuperberg, G. R. (2021). Word predictability effects are linear, not logarithmic: Implications for probabilistic models of sentence comprehension. *Journal of Memory and Language*, 116:104174.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Bunzeck, B. and Zarrieß, S. (2023). GPT-wee: Effective pre-training for downsized language models. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Cameron-Faulkner, T., Lieven, E., and Tomasello, M. (2003). A construction based analysis of child directed speech. *Cognitive science*, 27(6):843–873.

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., and Zhang, C. (2023). Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Carr, M. F., Jadhav, S. P., and Frank, L. M. (2011). Hippocampal replay in the awake state: a potential substrate for memory consolidation and retrieval. *Nature Neuroscience*, 14(2):147–153.

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*. Association for Computational Linguistics (ACL).

Chen, X. and Portelance, E. (2023). Grammar induction pretraining for language modeling in low resource contexts. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Cheng, Z., Aralikatte, R., Porada, I., Piano, C. S.-D., and Cheung, J. C. K. (2023). McGill BabyLM shared task submission: The effects of data formatting and structure biases. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Chobey, A., Smith, O., Wang, A., and Prasad, G. (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*. Association for Computational Linguistics (ACL).

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Chomsky, N. (1979). The logical structure of linguistic theory. *Synthese*, 40:317–352.

Ciaglia, F., Stella, M., and Kennington, C. (2023). Investigating preferential acquisition and attachment in early word learning through cognitive, visual and latent multiplex lexical networks. *Physica A: Statistical Mechanics and its Applications*, 612:128468.

Clark, A. and Lappin, S. (2010). *Linguistic Nativism and the Poverty of the Stimulus*. John Wiley & Sons, Hoboken, NJ.

Cochran, B. P., McDonald, J. L., and Parault, S. J. (1999). Too smart for their own good: The disadvantage of a superior processing capacity for adult language learners. *Journal of Memory and Language*, 41(1):30–58.

Coffman, K. G. and Odlyzko, A. M. (2002). Internet growth: Is there a "moore's law" for data traffic? *Handbook of massive data sets*, pages 47–93.

Cristia, A., Dupoux, E., Gurven, M., and Stieglitz, J. (2019). Child-directed speech is infrequent in a forager-farmer population: A time allocation study. *Child Development*, 90(3):759–773.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M. a., Senior, A., Tucker, P., Yang, K., Le, Q., and Ng, A. (2012). Large scale distributed deep networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

DeBenedetto, J. (2023). Byte-ranked curriculum learning for BabyLM strict-small shared task 2023. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Dettmers, T., Lewis, M., Belkada, Y., and Zettlemoyer, L. (2022). GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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.

Edman, L. and Bylinina, L. (2023). Too much information: Keeping training simple for BabyLMs. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Ehrlich, S. F. and Rayner, K. (1981). Contextual effects on word perception and eye movements during reading. *Journal of Verbal Learning and Verbal Behavior*, 20(6):641–655.

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

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

Foushee, R., Griffiths, T., and Srinivasan, M. (2016). Lexical complexity of child-directed and overheard speech: Implications for learning. In Papafragou, A., Grodner, D., Mirman, D., and Trueswell, J., editors, *Proceedings of the 38th Annual Meeting of the Cognitive Science Society, CogSci 2016*, Proceedings of the 38th Annual Meeting of the Cognitive Science Society, CogSci 2016, pages 1697–1702. The Cognitive Science Society. Publisher Copyright: © 2016 Proceedings of the 38th Annual Meeting of the Cognitive Science Society, CogSci 2016. All rights reserved.; 38th Annual Meeting of the Cognitive Science Society: Recognizing and Representing Events, CogSci 2016 ; Conference date: 10-08-2016 Through 13-08-2016.

Futrell, R., Qian, P., Gibson, E., Fedorenko, E., and Blank, I. (2019). Syntactic dependencies correspond to word pairs with high mutual information. In *Proceedings of the fifth international conference on dependency linguistics (depling, syntaxfest 2019)*, pages 3–13.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. (2020). The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Gauthier, J., Hu, J., Wilcox, E., Qian, P., and Levy, R. (2020). SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. Association for Computational Linguistics.

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Iii, H. D., and Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, 64(12):86–92.

Gerlach, M. and Font-Clos, F. (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.

Gilkerson, J., Richards, J. A., Warren, S. F., Montgomery, J. K., Greenwood, C. R., Kimbrough Oller, D., Hansen, J. H., and Paul, T. D. (2017). Mapping the

early language environment using all-day recordings and automated analysis. *American Journal of Speech-Language Pathology*, 26(2):248–265.

Goodkind, A. and Bicknell, K. (2018). Predictive power of word surprisal for reading times is a linear function of language model quality. In *Proceedings of the 8th Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2018)*, pages 10–18, Salt Lake City, Utah. Association for Computational Linguistics.

Goodman, J. C., Dale, P. S., and Li, P. (2008). Does frequency count? parental input and the acquisition of vocabulary. *Journal of child language*, 35(3):515–531.

Govindarajan, V. S., Rodriguez, J. D., Bostrom, K., and Mahowald, K. (2023). Lilbevo: Explorations of strategies for training language models in more humanlike ways. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Hale, J. (2001). A probabilistic Earley parser as a psycholinguistic model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Hart, B. and Risley, T. R. (1995). *Meaningful differences in the everyday experience of young American children*. Paul H. Brookes Publishing.

He, P., Liu, X., Gao, J., and Chen, W. (2021). DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Heath, S. B. (1983). *Ways with words: Language, life and work in communities and classrooms*. cambridge university Press.

Hesslow, D., Zanichelli, N., Notin, P., Poli, I., and Marks, D. (2022). Rita: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*.

Hill, F., Bordes, A., Chopra, S., and Weston, J. (2016). The Goldilocks principle: Reading children's books with explicit memory representations. In Bengio, Y. and LeCun, Y., editors, *4th International Conference on Learning Representations*.

Hoefler, T., Alistarh, D., Ben-Nun, T., Dryden, N., and Peste, A. (2021). Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *J. Mach. Learn. Res.*, 22(1).

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J. W., and Sifre, L. (2022). An empirical analysis of compute-optimal large language model training. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

Hong, X., Loáiciga, S., and Sayeed, A. B. (2023). A surprisal oracle for active curriculum language modeling. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Hoover, J. L., Du, W., Sordoni, A., and O'Donnell, T. (2021). Linguistic dependencies and statistical dependence. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2941–2963.

Hoover, J. L., Sonderegger, M., Piantadosi, S. T., and O'Donnell, T. J. (2022). The plausibility of sampling as an algorithmic theory of sentence processing. *PsyArXiv preprint*.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-efficient transfer learning for NLP. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Huang, K.-J., Arehalli, S., Kugemoto, M., Muxica, C., Prasad, G., Dillon, B., and Linzen, T. (2024). Large-scale benchmark yields no evidence that language model surprisal explains syntactic disambiguation difficulty. *Journal of Memory and Language*, 137:104510.

Huebner, P. A., Sulem, E., Cynthia, F., and Roth, D. (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.

Jumelet, J., Hanna, M., De Heer Kloots, M., Langedijk, A., Pouw, C., and van der Wal, O. (2023). ChapGTP, ILLC's attempt at raising a BabyLM: Improving data efficiency by automatic task formation. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. (2020). Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Kennedy, A. (1992). The spatial coding hypothesis. In *Eye movements and visual cognition: Scene perception and reading*, pages 379–396. Springer.

Kennedy, A., Brooks, R., Flynn, L.-A., and Prophet, C. (2003). The reader's spatial code. In *The Mind's Eye*, pages 193–212. Elsevier.

Kiela, D., Bartolo, M., Nie, Y., Kaushik, D., Geiger, A., Wu, Z., Vidgen, B., Prasad, G., Singh, A., Ringshia, P., Ma, Z., Thrush, T., Riedel, S., Waseem, Z., Stenetorp, P., Jia, R., Bansal, M., Potts, C., and Williams, A. (2021). Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Kuribayashi, T., Oseki, Y., Ito, T., Yoshida, R., Asahara, M., and Inui, K. (2021). Lower perplexity is not always human-like. 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 5203–5217, Online. Association for Computational Linguistics.

Lappin, S. and Shieber, S. M. (2007). Machine learning theory and practice as a source of insightinto universal grammar. *Journal of Linguistics*, 43(2):393–427.

Leong, C. T., Cheng, Y., Wang, J., Wang, J., and Li, W. (2023). Self-detoxifying language models via toxification reversal. In Bouamor, H., Pino, J., and Bali, K., editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4433–4449, Singapore. Association for Computational Linguistics.

Levy, R. (2008). Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Linzen, T. (2019). What can linguistics and deep learning contribute to each other? Response to Pater. *Language*, 95(1):e99–e108.

Linzen, T. (2020). How can we accelerate progress towards human-like linguistic generalization? In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Lison, P. and Tiedemann, J. (2016). OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In Calzolari, N., Choukri, K., Declerck, T., Goggi, S., Grobelnik, M., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

MacWhinney, B. (2000). *The CHILDES project: The database*, volume 2. Psychology Press.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Martinez, R. D., McGovern, H., Goriely, Z., Davis, C., Caines, A., Buttery, P., and Beinborn, L. (2023). Climb – curriculum learning for infant-inspired model building. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Marvin, R. and Linzen, T. (2018). Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

McCoy, R. T., Frank, R., and Linzen, T. (2018). Revisiting the poverty of the stimulus: Hierarchical generalization without a hierarchical bias in recurrent neural networks. In Rogers, T., Rau, M., Zhu, J., and Kalish, C., editors, *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, pages 2093—2098, Austin, TX.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. In Singh, A. and Zhu, J., editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

Meister, C., Pimentel, T., Haller, P., Jäger, L., Cotterell, R., and Levy, R. (2021). Revisiting the Uniform Information Density hypothesis. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 963–980, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mi, M. (2023). Mmi01 at the BabyLM challenge: Linguistically motivated curriculum learning for pretraining in low-resource settings. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Momen, O., Arps, D., and Kallmeyer, L. (2023). Increasing the performance of cognitively inspired data-efficient language models via implicit structure building. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Muennighoff, N., Rush, A., Barak, B., Le Scao, T., Tazi, N., Piktus, A., Pyysalo, S., Wolf, T., and Raffel, C. A. (2024). Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36.

Murty, S., Sharma, P., Andreas, J., and Manning, C. (2023). Grokking of hierarchical structure in vanilla transformers. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 439–448, Toronto, Canada. Association for Computational Linguistics.

Nangia, N. and Bowman, S. R. (2019). Human vs. muppet: A conservative estimate of human performance on the GLUE benchmark. In Korhonen, A., Traum, D., and Màrquez, L., editors, *Proceedings of the 57th Annual Meeting*

of the Association for Computational Linguistics, pages 4566–4575, Florence, Italy. Association for Computational Linguistics.

Newport, E. L. (1988). Constraints on learning and their role in language acquisition: Studies of the acquisition of american sign language. *Language Sciences*, 10(1):147–172.

Nguyen, T. Q. and Salazar, J. (2019). Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.

Oba, M., Haga, A., Fukatsu, A., and Oseki, Y. (2023). CoNLL shared task BabyLM challenge: Curriculum learning based on sentence complexity approximating language acquisition. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Ochs, E. (1982). Talking to children in western samoa. *Language in society*, 11(1):77–104.

Oh, B.-D. and Schuler, W. (2023). Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times? *Transactions of the Association for Computational Linguistics*, 11:336–350.

Oh, B.-D., Yue, S., and Schuler, W. (2024). Frequency explains the inverse correlation of large language models' size, training data amount, and surprisal's fit to reading times. *arXiv preprint arXiv:2402.02255*.

Opper, M., Morrison, J., and Siddharth, N. (2023). On the effect of curriculum learning with developmental data for grammar acquisition. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Pannitto, L. and Herbelot, A. (2020). Recurrent babbling: evaluating the acquisition of grammar from limited input data. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 165–176, Online. Association for Computational Linguistics.

Pater, J. (2019). Generative linguistics and neural networks at 60: Foundation, friction, and fusion. *Language*, 95(1):e41–e74.

Perfors, A., Tenenbaum, J. B., and Regier, T. (2011). The learnability of abstract syntactic principles. *Cognition*, 118(3):306–338.

Pimentel, T., Meister, C., Wilcox, E. G., Levy, R., and Cotterell, R. (2023). On the effect of anticipation on reading times. *Transactions of the Association for Computational Linguistics*.

Portelance, E., Duan, Y., Frank, M. C., and Lupyan, G. (2023). Predicting age of acquisition for children's early vocabulary in five languages using language model surprisal. *Cognitive Science*.

Power, A., Burda, Y., Edwards, H., Babuschkin, I., and Misra, V. (2022). Grokking: Generalization beyond overfitting on small algorithmic datasets. *CoRR*, abs/2201.02177.

Proskurina, I., Metzler, G., and Velcin, J. (2023). Mini minds: Exploring Bebeshka and Zlata baby models. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Reali, F. and Christiansen, M. H. (2005). Uncovering the richness of the stimulus: Structure dependence and indirect statistical evidence. *Cognitive Science*, 29(6):1007–1028.

Rohde, D. L. and Plaut, D. C. (1999). Language acquisition in the absence of explicit negative evidence: How important is starting small? *Cognition*, 72(1):67–109.

Rowe, M. L. (2008). Child-directed speech: Relation to socioeconomic status, knowledge of child development and child vocabulary skill. *Journal of child language*, 35(1):185–205.

Rumelhart, D. E., McClelland, J. L., Group, P. R., et al. (1986). *Parallel distributed processing, volume 1: Explorations in the microstructure of cognition: Foundations*. The MIT press.

Samuel, D. (2023). Mean BERTs make erratic language teachers: the effectiveness of latent bootstrapping in low-resource settings. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Samuel, D., Kutuzov, A., Øvrelid, L., and Velldal, E. (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.

Schaller, R. R. (1997). Moore's law: past, present and future. *IEEE spectrum*, 34(6):52–59.

Sevilla, J., Heim, L., Ho, A., Besiroglu, T., Hobbhahn, M., and Villalobos, P. (2022). Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE.

Shain, C., Meister, C., Pimentel, T., Cotterell, R., and Levy, R. (2022). Large-scale evidence for logarithmic effects of word predictability on reading time. *PsyArXiv preprint*.

Shazeer, N. (2020). GLU variants improve transformer. *CoRR*, abs/2002.05202.

Shen, Y., Tay, Y., Zheng, C., Bahri, D., Metzler, D., and Courville, A. (2021). StructFormer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *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 7196–7209, Online. Association for Computational Linguistics.

Shleifer, S., Weston, J., and Ott, M. (2021). Normformer: Improved transformer pretraining with extra normalization. *CoRR*, abs/2110.09456.

Shneidman, L. A. and Goldin-Meadow, S. (2012). Language input and acquisition in a mayan village: How important is directed speech? *Developmental science*, 15(5):659–673.

Smith, N. J. and Levy, R. (2013). The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1):159–216.

Staub, A. (2015). The effect of lexical predictability on eye movements in reading: Critical review and theoretical interpretation. *Language and Linguistics Compass*, 9(8):311–327.

Stella, M., Beckage, N. M., and Brede, M. (2017). Multiplex lexical networks reveal patterns in early word acquisition in children. *Scientific Reports*, 7(1):46730.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C., and Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Sutton, R. (2019). The bitter lesson. *Incomplete Ideas (blog)*, 13(1):38.

Thoma, L., Weyers, I., Çano, E., Schweter, S., Mueller, J. L., and Roth, B. (2023). Cogmemlm: Human-like memory mechanisms improve performance and cognitive plausibility of LLMs. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Timiryasov, I. and Tastet, J.-L. (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*. Association for Computational Linguistics (ACL).

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Canton-Ferrer, C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Van Schijndel, M. and Linzen, T. (2021). Single-stage prediction models do not explain the magnitude of syntactic disambiguation difficulty. *Cognitive Science*, 45(6):e12988.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2019). SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (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.

Wang, B., Ping, W., Xiao, C., Xu, P., Patwary, M., Shoeybi, M., Li, B., Anandkumar, A., and Catanzaro, B. (2022). Exploring the limits of domain-adaptive training for detoxifying large-scale language models. *Advances in Neural Information Processing Systems*, 35:35811–35824.

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

Warstadt, A., Choshen, L., Mueller, A., Williams, A., Wilcox, E., and Zhuang, C.

(2023a). Call for papers - The BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *CoRR*, abs/2301.11796.

Warstadt, A., Mueller, A., Choshen, L., Wilcox, E., Zhuang, C., Ciro, J., Mosquera, R., Paranjabe, B., Williams, A., Linzen, T., and Cotterell, R. (2023b). Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In Warstadt, A., Mueller, A., Choshen, L., Wilcox, E., Zhuang, C., Ciro, J., Mosquera, R., Paranjabe, B., Williams, A., Linzen, T., and Cotterell, R., editors, *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Warstadt, A., Parrish, A., Liu, H., Mohananey, A., Peng, W., Wang, S.-F., and Bowman, S. R. (2020a). BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Warstadt, A., Zhang, Y., Li, X., Liu, H., and Bowman, S. R. (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.

Whang, O. (2023). The race to make A.I. smaller (and smarter). *The New York Times*. Accessed: 2024-03-19.

Wilcox, E., Vani, P., and Levy, R. (2021). A targeted assessment of incremental processing in neural language models and humans. 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 939–952.

Wilcox, E. G., Futrell, R., and Levy, R. (2023a). Using computational models to test syntactic learnability. *Linguistic Inquiry*, pages 1–44.

Wilcox, E. G., Gauthier, J., Hu, J., Qian, P., and Levy, R. (2020). On the predictive power of neural language models for human real-time comprehension behavior. In *Proceedings of the Cognitive Science Society*.

Wilcox, E. G., Pimentel, T., Meister, C., and Cotterell, R. (2024). An information-theoretic analysis of targeted regressions during reading. *Cognition*, 249:105765.

Wilcox, E. G., Pimentel, T., Meister, C., Cotterell, R., and Levy, R. (2023b). Testing the predictions of surprisal theory in 11 languages. *Transactions of the Association for Computational Linguistics*, 11.

Wolf, L., Hosseini, E. A., Tuckute, G., Kotar, K., Warstadt, A., Wilcox, E., and Regev, T. I. (2023). WhisBERT: Multimodal text-audio language modeling on 100m words. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Xiao, C., Hudson, G. T., and Al Moubayed, N. (2023). Towards more human-like language models based on contextualizer pretraining strategy. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Yang, Y., Sulem, E., Lee, I., and Roth, D. (2023). Penn & BGU BabyBERTa+ for strict-small BabyLM challenge. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Yedetore, A., Linzen, T., Frank, R., and McCoy, R. T. (2023). How poor is the stimulus? Evaluating hierarchical generalization in neural networks trained on child-directed speech. In Rogers, A., Boyd-Graber, J., and Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9370–9393, Toronto, Canada. Association for Computational Linguistics.

Yehudai, A., Carmeli, B., Mass, Y., Arviv, O., Mills, N., Toledo, A., Shnarch, E., and Choshen, L. (2024). Genie: Achieving human parity in content-grounded datasets generation. *arXiv preprint arXiv:2401.14367*.

Zafrir, O., Boudoukh, G., Izsak, P., and Wasserblat, M. (2019). Q8bert: Quantized 8bit bert. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, pages 36–39, Los Alamitos, CA, USA. IEEE Computer Society.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. (2022). Scaling vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12104–12113.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M. T., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., and Zettlemoyer, L. (2022). OPT: Open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Zhang, Z., Yang, H., Ma, B., Rügamer, D., and Nie, E. (2023). Baby's CoThought: Leveraging large language models for enhanced reasoning in compact models. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Zhao, X., Wang, T., Osborn, S., and Rios, A. (2023). BabyStories: Can reinforcement learning teach baby language models to write better stories? In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Çağatan, Ö". V. (2023). ToddlerBERTa: Exploiting BabyBERTa for grammar learning and language understanding. In *Proceedings of the BabyLM Challenge*. Association for Computational Linguistics (ACL).

Appendix A. Common Approaches

The BabyLM Challenge received over 30 submissions. To streamline discussion, we taxonomize each submission according to the methods it employed.

Curriculum learning. This approach entails sorting the training data with respect to some complexity metric(s). It was the most popular approach, with 13 teams (41.9%) attempting some variant of curriculum learning. However, our meta-analysis found curriculum learning to be not very effective, with the majority of submissions that used it making only marginal gains over baseline models. That being said, they did explore a large space of possible curricula, for example: ranking sentences by average surprisal (Chobey et al., 2023; Hong et al., 2023), lexical frequency (Borazjanizadeh, 2023; Martinez et al., 2023), length (DeBenedetto, 2023; Edman and Bylinina, 2023), and syntactic complexity (Mi, 2023; Oba et al., 2023; Bunzeck and Zarrieß, 2023); sorting entire datasets by difficulty (Opper et al., 2023; Martinez et al., 2023; Xiao et al., 2023); gradually increasing vocabulary size (Thoma et al., 2023; Edman and Bylinina, 2023); and gradually increasing the difficulty of the training objective (Martinez et al., 2023). We discuss curriculum learning in greater detail in section 6.5.2.

Teacher–student or auxiliary model. In this setup an "teacher" model guides the learning dynamics of a target model. According to our rules, this was permissible as long as any auxiliary models were trained on the BabyLM corpus. Submissions that used this approach included Samuel (2023) and Berend (2023), and Timiryasov and Tastet (2023). Some of the notable submissions in this category used auxiliary models to select appropriate training examples for a curriculum (Chobey et al., 2023; Hong et al., 2023), or trained a reward model to use for reinforcement learning (Zhao et al., 2023).

Data preprocessing. Many submissions modified the format of the pretraining corpus. When controlled comparisons were performed, these preprocessing steps often led to improvements. Submissions in this category included (Govindarajan et al., 2023; Cheng et al., 2023; Edman and Bylinina, 2023). Among the more unique approaches in this space was the model submitted in (Zhang et al., 2023) (Baby's CoThought), which used an LLM to reformat unrelated sentences from the corpus into coherent paragraphs.

Hyperparameter tuning and model scaling. Many submissions performed extensive hyperparameter searches, producing combinations of hyperparameters that

work well on smaller datasets, leaving the underlying model architecture unchanged. While extensive hyperparameter searching can be expensive and challenging when scaling up to full-sized pretraining, in our limited data regime, such searches are much more tractable. Some hyperparameter changes that were found to result in improvements included reducing context length and training for more epochs or long epochs with data augmentation (Jumelet et al., 2023; Bhardwaj et al., 2023; Yang et al., 2023; Xiao et al., 2023; Samuel, 2023; Charpentier and Samuel, 2023). However, results are mixed when modifying model size: some participants achieved better results when increasing model sizes (Çağatan, 2023), while others were able to perform well when using very small models (Proskurina et al., 2023).

Multimodal learning. Multimodal learning was one of the directions where we expected the most interest and the most submissions, however, we received few submissions in this area, and the multimodal submissions did not reliably achieve high overall accuracy. One submission used music (Govindarajan et al., 2023), another used vision and language data (Amariucai and Warstadt, 2023), a third explored text-and-audio (Wolf et al., 2023), and a fourth incorporated text-and-image data and lexical sensorimotor data as part of the embedding process using multiplex networks (Stella et al., 2017; Ciaglia et al., 2023). Music training produced minor improvements on some subtasks, while the vision-and-language system marginally improved over the baselines in the *Strict-Small* track. The multiplex network did not produce performance gains, though it did allow the participants to reduce the number of parameters while preserving performance relative to the baselines.

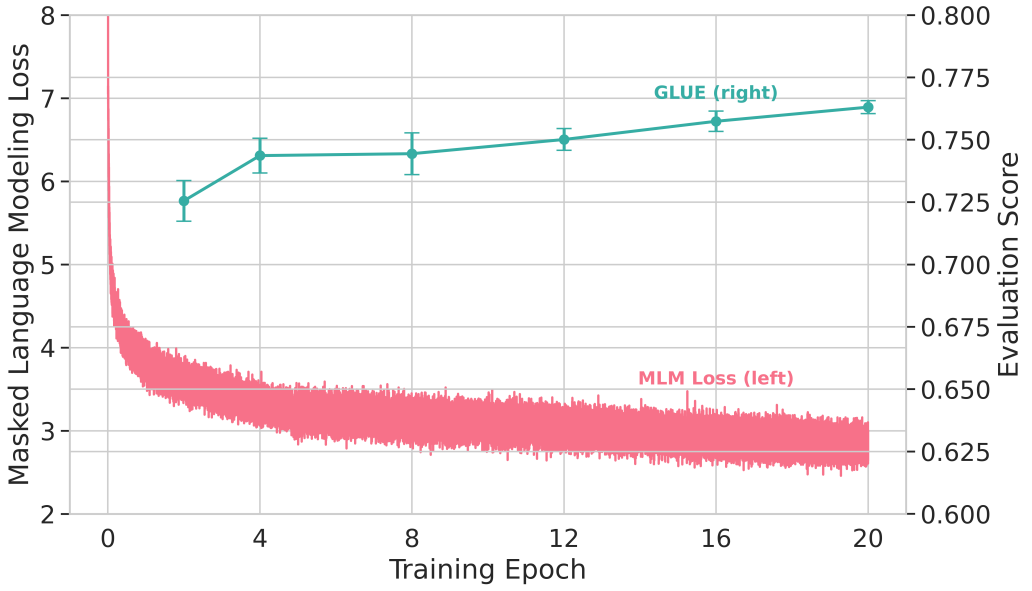
Architecture modifications. Submissions in this category implemented foundational changes to their underlying architecture that went above and beyond tuning pre-existing hyperparameters. As we will discuss in section §6.5.2, some of the most successful models involved this approach: Charpentier and Samuel (2023) added additional connections between network layers, by adding a weighted sum over the outputs of all previous layers. This architecture was based on a LTG-BERT backbone. Momen et al. (2023) used the relatively novel StructFormer architecture (Shen et al., 2021), which encourages tree-structured representations of inputs. Overall, the success of these models validated the mission of the challenge—the scaled-down data setting enabled participants to explore new and untested architectural choices, which proved to be ultimately effective.

Training objectives. Some submissions trained language models using a mixture of both a language modeling objective and some other objective. Martinez et al. (2023) simplified the masked language modeling objective by coarse-graining the output

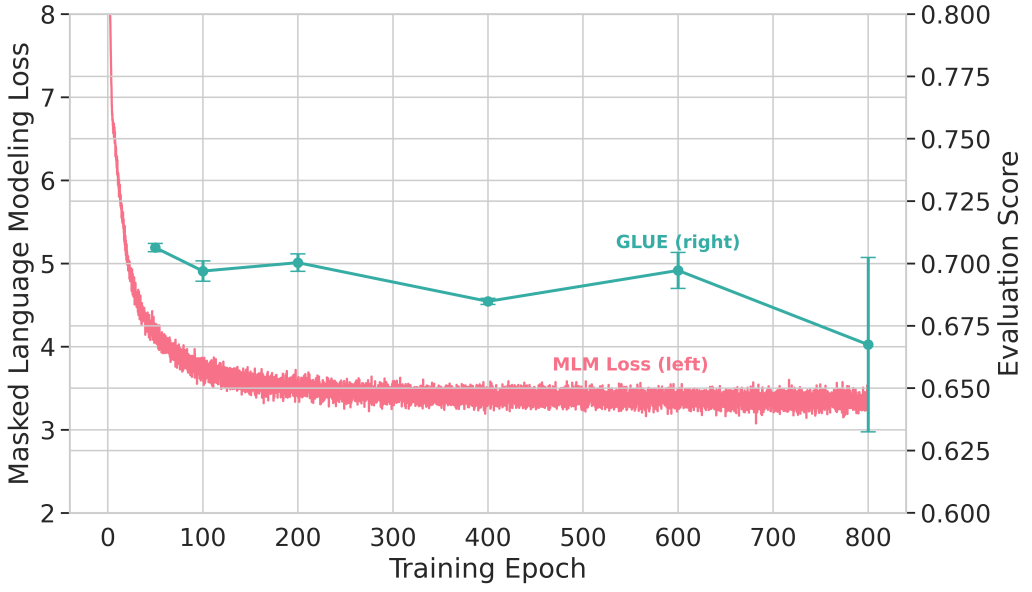
classes, with little effect. Govindarajan et al. (2023) achieved improvements on specific BLiMP subtasks by modifying the masking procedure to preferentially mask specific words thought to be relevant to a particular phenomenon tested by BLiMP.

Linguistic bias. Some submissions tried to impart human linguistic biases to models. Such approaches include curriculum learning based on linguistically motivated data sorting methods, as well as architectures that encourage hierarchical analyses of inputs. Chen and Portelance (2023). Finally Thoma et al. (2023) iteratively updated the vocabulary of the model based on word simplicity measures, which were motivated by human age-of-acquisition analyses.

Data augmentation. Some approaches used models to produce novel data, which were then incorporated in the training regime of a final model. One such model included the Contextualizer (Xiao et al., 2023) (see §6.5.2). In addition, Jumelet et al. (2023) used regular expressions to generate question-answer pairs given the BabyLM training data, and Zhao et al. (2023) used an LLM to generate text merging disparate sentences into cohesive paragraphs.



(a) *Strict*, 20 epochs



(b) *Strict-Small*, 800 epochs

Figure A.8: Training curves and GLUE evaluation scores for *Strict* and *Strict-Small* LTG-BERT. All losses and scores are averaged over 3 random seeds. GLUE "Evaluation Score" is an average over all task-specific metrics (typically accuracy or F1-score). GLUE performance for *Strict-Small* LTG-BERT declines after training for 50 epochs. The Pearson correlations between training loss and GLUE performance are -0.97 and 0.61 for *Strict* and *Strict-Small* respectively.

Appendix B. LTG-BERT reproduction details

We closely approximate the LTG-BERT results while training for 20 epochs on *Strict*, as opposed to 500 (Charpentier and Samuel, 2023). Mechanistically speaking, this finding says that LTG-BERT can be trained with a smaller batch size on fairly reasonable compute. We list our hyperparameters, along with the hyperparameters reported by the authors, below. Our ELC-BERT reproduction uses the same hyperparameters as *Strict* [R]. All training runs were done on 4 NVIDIA RTX8000 GPUs.

Hyperparameter	*Strict*	*Strict* [R]	*Strict-Small*	*Strict-Small* [R]
Number of parameters	98M	98M	24M	24M
Number of layers	12	12	12	12
Hidden size	768	768	384	384
FF intermediate size	2 048	2 048	1 024	1 024
Vocabulary size	16 384	16 384	6 144	6 144
Attention heads	12	12	6	6
Hidden dropout	0.1	0.1	0.1	0.1
Attention dropout	0.1	0.1	0.1	0.1
Training steps	15 625	**110 000**	31 250	**32 000**
Batch size	32 768	**256**	8 096	**2048**
Initial Sequence length	128	128	128	128
Final Sequence length	512	**128**	512	**128**
Warmup ratio	1.6%	1.6%	1.6%	1.6%
Initial learning rate	0.01	**3e-3**	0.005	0.005
Final learning rate	0.001	0.00141	0.005	0.005
Learning rate scheduler	cosine	cosine	cosine	cosine
Weight decay	0.1	0.1	0.4	0.4
Layer norm ϵ	1e-7	1e-7	1e-7	1e-7
Optimizer	LAMB	LAMB	LAMB	LAMB
LAMB ϵ	1e-6	1e-6	1e-6	1e-6
LAMB β_1	0.9	0.9	0.9	0.9
LAMB β_2	0.98	0.98	0.98	0.98
Gradient clipping	2.0	2.0	2.0	2.0

Table B.4: Pretraining hyperparameters. Differences between our training runs (labeled "[R]") and the original are bolded.