A Morpheme-Aware Child-Inspired Language Model

Necva Bölücü
CSIRO Data61, Australia
necva.bolucu@csiro.au

Burcu Can
University of Stirling, UK
burcu.can@stir.ac.uk

Abstract

Most tokenization methods in language models rely on subword units that lack explicit linguistic correspondence. In this work, we investigate the impact of using morpheme-based tokens in a small language model, comparing them to the widely used frequency-based method, BPE. We apply the morpheme-based tokenization method to both 10-million and 100-million word datasets from the BabyLM Challenge. Our results show that using a morphological tokenizer improves EWoK (basic world knowledge) performance by around 20% and entity tracking by around 40%, highlighting the impact of morphological information in developing smaller language models. We also apply curriculum learning, in which morphological information is gradually introduced during training, mirroring the vocabulary-building stage in infants that precedes morphological processing. The results are consistent with previous research: curriculum learning yields slight improvements for some tasks, but performance degradation in others.

1 Introduction

Large language models (LLMs) have substantially transformed the Natural Language Processing (NLP) domain (Brown et al., 2020). These models leverage vast datasets during pre-training to achieve state-of-the-art performance (Chang et al., 2024). For instance, earlier models, such as GPT-2, were trained on approximately 200 billion tokens (Radford et al., 2019), whereas more recent models, like Llama 3.1, have increased this requirement to over 15 trillion tokens (Grattafiori et al., 2024)[1]. This exponential increase in pre-training data demands highlights the resource-intensive nature of LLMs. Consequently, pre-training such models in low-resource environments poses significant challenges.

In stark contrast, human teenagers master language with exposure to just 100 million words over their whole lifetime (Warstadt et al., 2020a), highlighting a remarkable efficiency gap between human language learning and training LLMs. Therefore, emulating human language acquisition in LLMs could drastically reduce data requirements, making LLMs more viable and effective in resource-constrained settings (Warstadt et al., 2023).

The BabyLM Challenge[2], organized over the past two years (Warstadt et al., 2023; Hu et al., 2024), aims to develop more human-like, data-efficient approaches. To this end, it provides curated child-directed datasets that approximate both the quantity and quality of linguistic exposure experienced by children. These datasets form the basis of a controlled training environment designed to mimic the conditions of early language learning (Capone et al., 2024b). By focusing on such constrained input, the BabyLM Challenge promotes research into models that more closely reflect human-like learning trajectories under limited data regimes (Warstadt et al., 2023; Hu et al., 2024).

In this work, we introduce a morpheme-aware approach where the tokenizer simply splits words into morphologically meaningful units, unlike the other tokenizer methods such as BPE, WordPiece, or SentencePiece (Devlin et al., 2019; Kudo and Richardson, 2018a). This is inspired by child language acquisition, where the vocabulary building stage is followed by morphological and syntactic learning, where relationships between different word forms and words are learned later in the language acquisition (Tomasello, 2003; Clark, 2016).

In addition, we further investigate curriculum learning, in which morphological units are gradually introduced during training. This idea is in-

[1] OpenAI has not disclosed GPT-4's training data, but estimates suggest it was trained on over 13 trillion tokens.

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

November 8, 2025 ©2025 Association for Computational Linguistics

spired by child-directed language, where rephrasing is extensively used by employing different morphological forms of the same word in various phrases, and even by emphasizing the bare forms of nouns (i.e. stems) separately. While this approach is especially important for morphologically rich languages, we nonetheless examine it in the context of English, despite its relatively limited morphological complexity.

Our results show that morphological information significantly impacts language models. In particular, our EWoK and entity tracking scores are substantially higher than those obtained with a BPE tokenizer. These results are somewhat surprising, as EWoK measures basic world knowledge rather than a linguistic task. However, the substantial increase in entity tracking aligns closely with the linguistic nature of the task. Curriculum learning positively affects all tasks when using the GPT-BERT architecture (Charpentier and Samuel, 2024), whereas it degrades performance on BLIMP and BLIMP Supplement under the GPT-2 configuration. This is broadly consistent with prior research on curriculum learning (Capone et al., 2024a; Hong et al., 2023), which reports only modest improvements in language model performance.

2 Related Work

Here, we review related work on both tokenization methods and curriculum learning applied to small language models.

Tokenization methods in Small LMs: Bunzeck et al. (2024) use grapheme-based and character-based tokenization along with two different models: grapheme-llama and phoneme-llama. In the phoneme model, they convert the dataset into their phoneme representations, which drastically reduces the vocabulary size. Although the grapheme-based model outperforms the phoneme-based model, the results show that the model can learn the structure of language using only characters as tokens. Analogously, Goriely et al. (2024) use phoneme representations of the dataset. Although the results are slightly lower in language understanding tasks, such phoneme representations have practical advantages, such as in multilingual language modeling.

To our knowledge, this paper is the first to explore morpheme-based tokenization in small language models.

Curriculum Learning Several previous BabyLM Challenge submissions have explored curriculum learning as a strategy to enhance data efficiency and developmental plausibility in language modelling. Diehl Martinez et al. (2023) introduced a curriculum learning framework inspired by infant cognitive development, organizing data to reflect the incremental complexity faced by human learners. Similarly, DeBenedetto (2023) proposed a simple, computationally efficient method for sequencing training data by byte-level difficulty, demonstrating modest gains over random baselines. Oba et al. (2023) approximated natural language acquisition by reordering sentences according to syntactic and lexical complexity, reflecting stages in child language development. Building on the same idea, Hong et al. (2023) used model-based surprisal estimates to dynamically select training examples, aiming to optimize learning trajectories through adaptive data exposure.

In 2024, several approaches continued this trend with more refined techniques. ConcreteGPT (Capone et al., 2024a) implemented a curriculum based on lexical concreteness, training models to first acquire concrete vocabulary before progressing to more abstract terms, thereby mirroring patterns in early word learning.

To the best of our knowledge, no prior small language model has investigated morpheme-based curriculum learning, drawing inspiration from child language acquisition in which vocabulary development precedes the acquisition of morphology and syntax.

3 Methodology

In this study, we investigate morphologically informed tokenization and its impact on language modeling in data-limited contexts, with a particular focus on the BabyLM setting. We employ morpheme-aware tokenization alongside curriculum learning, exploring how these strategies can improve both the efficiency and linguistic generalization of models trained on small corpora. Our approach centers on two key components: (1) the tokenization method and (2) the training regime, with an emphasis on mimicking the stages of early human language development.

3.1 Tokenization Strategies

We compare three tokenization approaches with varying degrees of linguistic awareness: (1) a

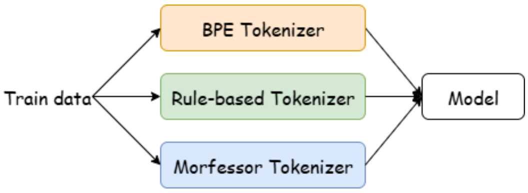

Figure 1: Tokenizers

BPE tokenizer; (2) a rule-based morphological tokenizer; and (3) an unsupervised morphological tokenizer (Morfessor). An overview of the language model, along with the selected tokenizers, is provided in Figure 1.

Byte-pair Encoding (BPE) BPE is a widely used subword tokenization method that segments words based on the frequency of symbol pairs (Gage, 1994; Sennrich et al., 2016). While effective for vocabulary compression and handling out-of-vocabulary words, BPE is agnostic to morphological structure. We include BPE as a standard baseline to evaluate whether morphology-aware tokenizers provide superior advantages in the low-resource BabyLM setting.

Rule-based Tokenizer (Simple) To explicitly incorporate morphological information, we develop a simple rule-based tokenizer that segments words using a predefined list of common English prefixes and suffixes (e.g., 'in', 'un', 'ed', 'ing', 's', etc.). The tokenizer iteratively strips recognized suffixes from the ends of words and prefixes from the beginnings. For example, the word *undoing* is segmented into *un* + *do* + *ing* by identifying 'un' as a prefix, and 'ing' as a suffix using the pre-defined morpheme list. Words shorter than four characters are excluded to reduce oversegmentation. This tokenizer is inspired by early stages of human vocabulary learning, where affix awareness emerges before complex syntactic structures (Tomasello, 2003; Clark, 2016). The method is deterministic, lightweight, and interpretable, making it especially suitable for low-resource conditions. However, it is language-specific and requires a predefined list of morphemes for each target language.

Unsupervised Tokenizer (Morfessor) As a second method for morpheme-based tokenization, we also use the Morfessor (Virpioja et al., 2013), which is an unsupervised morphological analyzer. Unlike BPE, Morfessor produces linguistically plausible segmentations, offering a data-driven but morphology-aware alternative that aligns with our hypothesis about the importance of structured vocabulary building. Moreover, unlike the rule-based morphological tokenizer, it is language-agnostic and can be trained on any language using only a raw corpus.

Table 1 presents sample tokenization outputs for words ranging from morphologically simple to complex, highlighting the differences in segmentation strategies among various tokenizers. As seen, BPE tends to oversegment words depending on their frequency in the dataset, whereas Morfessor and the Simple tokenizer tend to produce longer tokens that better align with the morphemes of the language. However, they remain prone to errors, though they are still better aligned with the morphological structure of words.

3.2 Data

We use the official datasets provided by the BabyLM Challenge: the 10M (Strict-Small) and 100M (Strict) word text-only datasets. These are drawn from a variety of sources, including BNC (Burnard, 2007), CHILDES (Pye, 1994), children's books from Project Gutenberg (Gerlach and Font-Clos, 2020), Simple English Wikipedia, Switchboard (Stolcke et al., 2000), and OpenSubtitles (Lison and Tiedemann, 2016).

We clean the datasets using the cleaning script[3] provided by Timiryasov and Tastet (2023) before training the models.

3.3 Training

We adopt two architectures in our experiments: GPT-2 (Charpentier et al., 2025) and GPT-BERT (Charpentier and Samuel, 2024). Under two architectures, we follow two training approaches in our experiments.

The first training approach is merely built on one of the tokenization methods described above (i.e. BPE, rule-based morphological tokenizer, unsupervised tokenizer), and it involves only one training phase. In the second training approach, we use curriculum learning where the morphological structure of language is gradually introduced during training. In curriculum learning, the first phase corresponds to the vocabulary-building stage in babies, whereas the second phase corresponds to building morphology and syntax, building on top of the vocabulary learned in the first phase.

[3] https://huggingface.co/timinar/baby-llama-58m/blob/main/mrclean.py

Word	BPE	Simple	Morfessor
run	r, un	run	run
dog	d, og	dog	dog
redo	red, o	re, do	re, do
cats	c, ats	cats	cats
jumping	j, ump, ing	jump, ing	jump, ing
played	play, ed	play, ed	played
unhappy	un, happy	un, happy	un, happy
happiness	ha, pp, iness	happi, ness	happiness
friendliness	friend, l, iness	friendli, ness	friendliness
undeniable	un, deniable	un, deniable	undeniable
counterattack	counter, att, ack	counterattack	counter, attack
unbelievably	un, bel, ie, v, ably	un, believab, ly	unbeliev, ably
reconsideration	re, c, ons, ider, ation	re, considera, tion	re, consideration
misunderstanding	m, is, under, standing	misunderstand, ing	misunderstand, ing

Table 1: Comparison of tokenization outputs for selected words by BPE, Simple, and Morfessor tokenizers.

3.4 Evaluation

We evaluate our models through the BabyLM evaluation pipeline (Charpentier et al., 2025). This pipeline consists of six tasks that collectively probe different dimensions of linguistic and cognitive ability.

BLiMP (Warstadt et al., 2020b) measures grammatical knowledge through minimal pair judgments. It consists of minimal pairs of sentences where one is grammatically well-formed and the other is not. EWoK (Ivanova et al., 2024) evaluates basic world knowledge. (Super)GLUE (Wang et al., 2018, 2019) tests general natural language understanding across multiple benchmarks. Entity Tracking (Kim and Schuster, 2023) assesses a model's ability to maintain reference to entities across discourse. Reading (de Varda et al., 2024) evaluates cloze-style reading comprehension. Finally, WUG (Hofmann et al., 2025b) examines the ability of a model to generalize to novel word forms, reflecting morphological productivity. Together, these tasks provide a comprehensive evaluation of models in terms of syntax, semantics, discourse, and generalization, aligning with the developmental plausibility focus of the BabyLM Challenge.

The hidden tasks cover diverse aspects of linguistic competence. WUG_PAST (Weissweiler et al., 2023) tests morphological generalization by correlating model-predicted past tense forms of nonce words with human responses, while WUG_ADJ (Hofmann et al., 2025a) applies the same correlation-based evaluation to adjective nominalization (-ity vs. -ness). COMPS (Misra et al., 2023) probes property inheritance using minimal pairs with nonce concepts, rewarding higher probability for correct sentences. The AoA Benchmark (Chang and Bergen, 2022) tracks surprisal across training to fit learning curves and correlates model-derived acquisition ages with human norms from the MacArthur–Bates CDI[4].

Evaluation metrics We report only zero-shot experiment results on BLiMP, BLiMP Supplement, EWoK, Entity Tracking, and WUG. For reading tasks, we evaluate performance using the coefficient of determination (R^2): Eye Tracking is assessed without spillover, while Self-paced Reading is evaluated with a one-word spillover.

4 Experiments & Results

We use two language model architectures for training the models: GPT-2 (Radford et al., 2019)[5] and GPT-BERT (Charpentier and Samuel, 2024)[6], the winner of the BabyLM 2024. We compare the results with the official results of baselines in BabyLM 2024. The baselines are also based on GPT-2 and GPT-BERT, all using BPE as the tokenizer. GPT-BERT includes two variants, trained with causal language modeling (CLM) and masked next token prediction (MNTP), respectively.

Tokenizer For all tokenizers, we train them on the training corpus with a vocabulary size of $2^{13} = 8192$ in all configurations.

GPT-2 Configuration We adopt the GPT-2 small architecture (Radford et al., 2019), consisting of 12 transformer decoder layers with 12 attention heads,

[4] https://wordbank.stanford.edu/

[5] https://github.com/momergul/babylm-gpt2-baseline

[6] https://github.com/ltgoslo/gpt-bert/

STRICT-SMALL track (10M words)

Model	Tokenizer	BLiMP	BLiMP Supplement	EWoK	Eye tracking	Self-paced Reading	Entity Tracking	WUG
GPT-2	BPE	65.77	62.40	49.82	0.73	0.03	21.93	52.00
GPT-2	SimpleTokenizer	53.04	44.40	53.55	0.74	0.08	40.66	100.00
GPT-2	Morfessor	65.10	49.20	68.45	0.08	0.12	59.65	100.00
GPT-2 (curriculum)	Morfessor	63.19	48.80	69.64	0.09	0.26	59.82	100.00
GPT-BERT	BPE	68.70	61.50	50.40	6.20	**4.45**	25.30	44.50
GPT-BERT	SimpleTokenizer	56.45	49.18	53.18	0.91	0.05	42.18	100.00
GPT-BERT	Morfessor	69.10	50.08	70.01	0.09	0.06	62.17	100.00
GPT-BERT (curriculum)	Morfessor	**72.10**	52.12	**71.15**	0.12	0.36	**63.25**	100
babylm-baseline-10m-gpt2	BPE	66.36	57.07	49.90	8.66	4.34	13.9	52.5
babylm-baseline-10m-gpt-bert-causal	BPE	65.22	59.49	49.47	**9.52**	3.44	30.60	68.00
babylm-baseline-10m-gpt-bert-mntp	BPE	70.36	**63.71**	49.95	9.40	3.37	40.02	57.5

STRICT track (100M words)

Model	Tokenizer	BLiMP	BLiMP Supplement	EWoK	Eye tracking	Self-paced Reading	Entity Tracking	WUG
GPT-2	BPE	75.24	62.80	51.00	2.70	0.43	25.48	47.00
GPT-2	SimpleTokenizer	71.10	48.56	59.17	0.76	0.32	63.10	100.00
GPT-2	Morfessor	64.60	55.20	67.45	0.81	0.28	67.45	100.00
GPT-2 (curriculum)	Morfessor	63.12	49.60	67.82	0.69	0.32	49.47	100.00
GPT-BERT	BPE	79.60	42.60	52.00	6.20	3.05	25.30	45.00
GPT-BERT	SimpleTokenizer	69.18	58.17	69.18	1.05	0.35	67.56	100.00
GPT-BERT	Morfessor	70.12	56.18	69.56	0.98	0.32	**68.48**	100.00
GPT-BERT (curriculum))	Morfessor	73.36	58.43	**71.15**	1.09	0.46	60.21	100.00
babylm-baseline-100m-gpt2	BPE	74.88	63.32	51.67	7.89	3.18	31.51	35.5
babylm-baseline-10m-gpt-bert-causal	BPE	74.56	63.63	51.57	8.80	3.30	30.82	59.00
babylm-baseline-10m-gpt-bert-mntp	BPE	**80.75**	**75.34**	51.77	**9.34**	**3.34**	41.15	55.00

Table 2: Performance of different models across multiple evaluation benchmarks.

a hidden size of 768. The model uses standard initialization (`initializer_range`=0.02) and layer normalization ($\epsilon = 1e^{-5}$). We train for 200k steps with a batch size of 16, using Adam with a learning rate of 5e-5 and 2k warm-up steps. Weight decay is set to zero. The same configuration is used for both strict (100M) and strict-small (10M) data. This configuration contains approximately 124M parameters.

GPT-BERT Configuration We adopt the GPT-BERT architecture (Charpentier and Samuel, 2024) which was the winner of BabyLM 2024. Our implementation follows the configuration reported in the original study, except for the vocabulary size, consisting of 12 transformer layers with a hidden size of 768, weight decay of 0.1, and hidden and attention dropout of 0.1. For the *strict* data (100M), we use 12 attention heads, resulting in approximately 119M parameters, while for the *strict-small* data (10M), we use 6 attention heads with a hidden size of 384, yielding about 30M parameters.

To further limit the computational cost of training, we restrict the context length of the model to 512 tokens in all experiments. All experiments have been carried out locally on one Nvidia H100 GPU.

4.1 Zero-shot Experiments

Table 2 reports results for GPT-2 and GPT-BERT with BPE, SimpleTokenizer, and Morfessor, under both single-stage training and curriculum learning, for the Strict-Small track (10M words) and the Strict track (100M words). Morpheme-based tokenization shows a clear impact on zero-shot tasks, particularly in cognitively demanding settings such as Entity Tracking and EWoK. Models using Morfessor consistently outperform those with BPE or SimpleTokenizer on these benchmarks, often by a substantial margin (e.g., over 20% in EWoK and nearly 40% in Entity Tracking). This improvement likely stems from Morfessor's linguistically informed segmentation, which aligns subword units with meaningful morphological boundaries. By preserving semantic units within words, Morfessor enables the model to better capture entity consistency and relationships, enhancing its ability to track entities across discourse and reason about their attributes. These findings highlight the advantages of morphology-aware tokenization in low-resource settings where semantic richness and structural sensitivity are essential. Interestingly, while BLiMP scores are comparable between BPE and Morfessor, morpheme-based tokenizers perform substantially worse on BLiMP Supplement.

Curriculum learning yields slight improvements across all scores in the GPT-BERT configuration, but results in minor performance degradation with GPT-2. This suggests that the training strategy does not have a uniform effect on performance, but rather interacts differently with specific architectures. The modest gains observed with curriculum learning are consistent with prior research, which has generally reported small improvements from multi-stage training using data blocks of varying difficulty (Capone et al., 2024a; Hong et al., 2023).

5 Conclusion

We showed the effectiveness of using a morpheme-based tokenizer in low resource settings to train a baby language model. Our results show that a morpheme-based tokenizer outperforms BPE for some tasks, such as EWoK and entity tracking by a substantial margin.

We only used GPT-2 and GPT-BERT for the backbone architecture. The results also show that the impact of a tokenizer can be quite different in different architectures. For example, we also investigated curriculum learning using the morphological complexity as the main criterion in a phased training, and the results are different in GPT-2 and GPT-BERT. The morpheme-based tokenizer improves all the scores, including BLIMP, BLIMP Supplement, EWoK, eye-tracking, and entity tracking, when used with the GPT-BERT architecture, whereas curriculum learning does not help as desired when used with the GPT-2 architecture.

Limitations

We showed the effectiveness of a morpheme-based tokenizer for English, a morphologically-poor language. This choice may have hindered the tokenizer's performance, and its application to a morphologically rich language, such as Turkish, could yield significantly different results. In the future, we aim to apply this method to morphologically rich languages in limited-resource settings.

Although we showed the superiority of a morpheme-based tokenizer over a count-based one like BPE, we did not compare it against other methods such as SentencePiece (Kudo and Richardson, 2018b), or character- and word-level tokenizers. Therefore, its relative performance remains to be determined.

Furthermore, our investigation of curriculum learning was limited to morphological complexity. We did not explore syntactic complexity, which, in child language acquisition, is integral to vocabulary building and follows morphological processing.

Ethics Statement

This study was conducted in accordance with ethical guidelines and regulations. We utilized natural speech data extracted from CHILDES (MacWhinney, 2000). This is an open-source corpus that archives natural speech between caregivers and their children. The data are archived without confidential information about the participants, as children are usually given pseudonyms. Following the ACL Policy on Publication Ethics, we used ChatGPT to assist in refining the wording.

Acknowledgements

We wish to acknowledge Tharindu Ranasinghe for stimulating discussions related to this research.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2024. Graphemes vs. phonemes: battling it out in character-based language models. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 54–64, Miami, FL, USA. Association for Computational Linguistics.

Lou Burnard. 2007. Reference guide for the British national corpus (XML Edition).

Luca Capone, Alessandro Bondielli, and Alessandro Lenci. 2024a. ConcreteGPT: A baby GPT-2 based on lexical concreteness and curriculum learning. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 189–196, Miami, FL, USA. Association for Computational Linguistics.

Luca Capone, Alice Suozzi, Gianluca Lebani, and Alessandro Lenci. 2024b. BaBIEs: A benchmark for the linguistic evaluation of Italian baby language models. In *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 157–170, Pisa, Italy. CEUR Workshop Proceedings.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word Acquisition in Neural Language Models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A Survey on Evaluation of Large Language Models. *ACM Trans. Intell. Syst. Technol.*, 15(3).

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

Lucas Georges Gabriel Charpentier and David Samuel. 2024. GPT or BERT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.

Eve V. Clark. 2016. *First Language Acquisition*, 3 edition. Cambridge University Press.

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

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Richard Diehl Martinez, Zébulon Goriely, Hope McGovern, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – curriculum learning for infant-inspired model building. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 112–127, Singapore. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

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

Zébulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. From babble to words: Pre-training language models on continuous streams of phonemes. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 37–53, Miami, FL, USA. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

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

Xudong Hong, Sharid Loáiciga, and Asad Sayeed. 2023. A surprisal oracle for active curriculum language modeling. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 259–268, Singapore. Association for Computational Linguistics.

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

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

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

Taku Kudo and John Richardson. 2018a. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018b. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Brian MacWhinney. 2000. *The CHILDES Project: Tools for analyzing talk. Third Edition*. Lawrence Erlbaum Associates.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual Minimal Pair Sentences for testing Robust Property Knowledge and its Inheritance in Pre-trained Language Models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Miyu Oba, Akari Haga, Akiyo Fukatsu, and Yohei Oseki. 2023. BabyLM challenge: Curriculum learning based on sentence complexity approximating language acquisition. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 290–297, Singapore. Association for Computational Linguistics.

Clifton Pye. 1994. The CHILDES project: Tools for analyzing talk.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

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

Inar Timiryasov and Jean-Loup Tastet. 2023. Baby Llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for Morfessor Baseline.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020a. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020b. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the Bugs in ChatGPT's Wugs: A Multilingual Investigation into the Morphological Capabilities of a Large Language Model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.