A Comparison of Elementary Baselines for BabyLM

Rareș Păpușoi and Sergiu Nisioi*
Human Language Technologies Research Center
Faculty of Mathematics and Computer Science
University of Bucharest
rares.papusoi@s.unibuc.ro
sergiu.nisioi@unibuc.ro

Abstract

This paper explores several simple baselines for the BabyLM Challenge, including random models, elementary frequency-based predictors, n-gram language models, LSTMs with various tokenizers (BPE, Unigram, SuperBPE), and GPT-BERT, the winning architecture from the previous BabyLM edition. Evaluation focuses on the BLiMP and BLiMP-Supplement benchmarks. Our experiments reveal that the STRICT-SMALL corpus can sometimes outperform STRICT, that performance is highly sensitive to tokenization, and that data efficiency plays a crucial role. Notably, a simple word-frequency baseline achieved unexpectedly high scores, which led us to identify an evaluation artifact in the pipeline: a system assigning identical sentence-level log-likelihoods to both sentences can attain maximal accuracy. The code for our experiments is available at https://github.com/rarese19/babylm_baselines

1 Introduction

The BabyLM Challenge targets sample-efficient pretraining on developmentally plausible text under strict data budgets (Charpentier et al., 2025). It provides text-only training splits capped at 10M (STRICT-SMALL) and 100M (STRICT) words, drawn from child-directed and conversational sources, with standardized evaluation (Charpentier et al., 2025). In this paper, we operate entirely within the text-only track and treat BabyLM as a fixed environment. Within this setting, we study how model family, tokenizer, and corpus influence grammatical competence under these constraints.

To train our models, we use the official STRICT and STRICT-SMALL corpora, as well as the BabyCosmoFine mixture (Charpentier and Samuel, 2024), which combines equal parts of the BabyLM subset, FineWeb, and Cosmopedia. Evaluation is conducted mostly on BLiMP (Benchmark of Linguistic Minimal Pairs) (Warstadt et al., 2020) and BLiMP-Supplement (Warstadt et al., 2023) from the 2024 evaluation pipeline.

Our results fill a gap in the BabyLM Challenge by providing a comparison between trivial baselines (e.g., random predictions, frequency-based models) and full language models. The contributions of this paper are summarized as follows:

- We present a controlled comparison of model families (n-gram language models, long short-term memory — LSTM, GPT-BERT) and tokenizers (byte-pair encoding, Unigram, SentencePiece, SuperBPE) under fixed BabyLM data budgets and corpora.
- We show that trivial lexical baselines (e.g., raw or Zipf-distributed word frequency) can perform surprisingly well.
- We identify and quantify an evaluator caveat: 22 BLiMP subtasks are permutation-equivalent, allowing order-insensitive systems to tie both sentences. The issue persists on the 2025 evaluation pipeline in a slightly different formulation.
- We establish a strong LSTM (Hochreiter and Schmidhuber, 1997) baseline with 39.2M parameters and analyze tokenizer sensitivity, showing that an 8K vocabulary size with the SuperBPE tokenizer achieves the best performance.
- We train a similar Masked Next Token Prediction model based on slightly altered GPT-BERT recipe (Charpentier and Samuel, 2024). The model achieves the best results and shows that the BabyCosmoFine corpus is better suited for training models on dialogue and question-answering tasks than STRICT-SMALL.

*Corresponding authors.

Proceedings of the First BabyLM Workshop, pages 218–225

Dataset	# Words	
	STRICT	STRICT-SMALL
CHILDES (MacWhinney, 2000)	29M	2.9M
British National Corpus (BNC)	8M	0.8M
Proj. Gutenberg (Gerlach and Font-Clos, 2020)	26M	2.6M
OpenSubtitles (Lison and Tiedemann, 2016)	20M	2.0M
Simple English Wikipedia	15M	1.5M
Switchboard Dialog (Stolcke et al., 2000)	1M	0.1M
Total	**100M**	**10M**

Table 1: Composition of the STRICT and STRICT-SMALL datasets used in BabyLM, adapted from Charpentier et al. (2025).

2 Method

2.1 Datasets

The BabyLM Challenge datasets have only one constraint: they must not surpass 100M words (Charpentier et al., 2025). The competition allows for custom-made datasets as long as they comply with the limitation (Hu et al., 2024; Charpentier et al., 2025). This paper focuses on the STRICT-SMALL track, with a few supplementary experiments conducted under the STRICT category. All experiments use text-only datasets, excluding multimodal and other tracks available in the competition.

The BabyLM Dataset is constructed from multiple text sources in order to create diversity in the language style and the content (Warstadt et al., 2023). It contains text similar to what a child is exposed to during the language acquisition process. The STRICT-SMALL dataset extracts 10% of the 100M words that make up the STRICT corpus and keeps the distribution from the different sources (Warstadt et al., 2023; Charpentier et al., 2025). The dataset's structure is described in Table 1.

The BabyCosmoFine Corpus is created to provide a wider source of information for knowledge extraction and to diversify the language. It consists of a portion of the BabyLM dataset, a portion of the FineWeb-Edu corpus (Penedo et al., 2024), and a portion of the synthetic dataset Cosmopedia (Ben Allal et al.). Each component contributes equally, in terms of quantity, to the overall composition of the corpus (Charpentier and Samuel, 2024).

2.2 Tokenization

We explore a variety of options for data tokenization. Our approach consists of four tokenization schemes, BPE (Gage, 1994; Sennrich et al., 2016), Unigram (Kudo, 2018), their SentencePiece (Kudo and Richardson, 2018) implementations, and SuperBPE (Liu et al., 2025). The tokenizers are trained on both the BabyLM and BabyCosmoFine corpora. BPE incrementally merges frequent adjacent characters to reach a target vocabulary (Sennrich et al., 2016), while Unigram fits a simple probabilistic model over a large possible set and eliminates low-probability sub-words (Kudo, 2018). SentencePiece is an open-source library that provides BPE and Unigram implementations and works directly on raw text without any language-specific pre-tokenization. It treats spaces as a dedicated symbol (e.g., "_"), therefore, segmentation is learned from character sequences instead of using whitespace-defined word boundaries (Kudo and Richardson, 2018).

SuperBPE extends Byte Pair Encoding (BPE) (Liu et al., 2025) with a second training stage that removes whitespace boundaries and learns **super-words** i.e., tokens that can span multiple words (e.g., *by the way*, *I am!*). In the first stage, a standard BPE vocabulary is learned; in the second stage, the learned tokens are re-merged without enforcing spaces as hard boundaries, enabling frequent multi-word expressions to be represented as single tokens. All our SuperBPE tokenizers follow the default configuration, combining 90% regular tokens and 10% super-words.

2.3 Evaluation

Models are evaluated on tasks designed to assess core linguistic abilities and generalization from limited data. The evaluation suite spans grammatical phenomena, general knowledge, information tracking, reading comprehension, and morphological derivation. In this work, we focus on the minimal-pair grammatical acceptability suite, which consists of pairs of nearly identical sentences where the goal is to prefer the grammatical one. These tasks target syntax, morphology, and semantics (e.g., subject–verb agreement, binding).

BLiMP (Benchmark of Linguistic Minimal Pairs) evaluates a model's grammatical competence using sentence pairs that differ in exactly one syntactic, morphological, or semantic feature. Each example contains two sentences (one grammatical and one ungrammatical) and the model must identify the grammatical one by assigning it a higher probability (Warstadt et al., 2020).

BLiMP-Supplement extends BLiMP with examples focused on dialogue and question constructions. This test set was first introduced in the initial edition of the BabyLM Challenge. Its structure mirrors that of BLiMP but targets linguistic phenomena characteristic of conversational language (Warstadt et al., 2023).

The scoring method is based on sentence-level log-likelihood. Autoregressive models are evaluated by summing token-level log-probabilities, whereas masked language models use pseudo log-likelihood, computed by masking each token in turn and summing the resulting log-probabilities. For each minimal pair, the higher-scoring sentence is considered correct, and accuracy is then aggregated across subtasks.

2.4 Approaches

We consider simple baselines, alongside the ones provided by the competition (Choshen et al., 2024; Charpentier et al., 2025), classical n-grams, a recurrent model, and transformers, all within the BabyLM constraints.[1]

The controlled-random baseline sets a target sentence log-probability $R \sim \mathcal{U}[-100, 0]$ and returns the same logit vector at every position, independent of the input. Let L be the number of scored tokens and V the vocabulary size. We choose a constant logit α so that each reference token has probability $p = \frac{e^\alpha}{e^\alpha + (V-1)}$, which yields a sentence score

$$L\big[\alpha - \ln\big(e^\alpha + V - 1\big)\big] = R \tag{1}$$

We solve for α via a short binary search and then output the same logit vector at every position, α on the reference token at that position and 0 on all others, independent of the input.

The word frequency baseline assigns a sentence score by summing each word's relative frequency in a reference corpus. We use a frequency table based on each sub-corpus. For a sentence x,

$$S(x) = \sum_{w \in x} \mathrm{freq}(w) \tag{2}$$

with

$$\mathrm{freq}(w) = \begin{cases} \dfrac{c(w)}{N}, & c(w) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where $c(w)$ is the corpus count of w and N is the total token count. This score ignores word order and context.

As an alternative, we use a Zipf-style score inspired by `wordfreq` (Speer, 2022; Van Heuven et al., 2014):

$$\mathrm{zipf}(w) = \begin{cases} 3 + \log_{10}\big(10^6 c(w)/N\big), & c(w) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The factor 10^6 scales to "per million" and the constant 3 keeps values positive.

N-gram language models are trained with KenLM (Heafield, 2011) for orders $n \in \{2, \ldots, 6\}$ on the STRICT and STRICT-SMALL corpora. Training uses sentence-boundary markers, and a single <unk> token for out-of-vocabulary items; the tokenization is done with the default configuration of KenLM. Models are compiled to KenLM binaries and evaluated by the BabyLM evaluation pipeline, identical to the neural models.

Long Short-Term Memory (LSTM) is a neural language model (Hochreiter and Schmidhuber, 1997) with 39.2 million parameters. We train the model on the STRICT and STRICT-SMALL splits, using the tokenizers described in Section 2.2 (BPE, Unigram, SuperBPE). Each configuration is trained for 10 epochs with identical hyperparameters across corpora, and no external text is used.

Transformer language models (Vaswani et al., 2017) are treated as standard modeling tools, with our focus placed primarily on training objectives and configurations. GPT-BERT, the winning model from the 2024 BabyLM Challenge, employs Masked Next Token Prediction (MNTP) alongside a causal language modeling objective (BehnamGhader et al.; Charpentier and Samuel, 2024). We follow the publicly released training recipe, with using the script designed for single GPU training.[2] The script schedules late-training changes at 70% and 90% of steps; due to resource constraints we apply only the $\geq 70\%$ change and

[1] The 2025 BabyLM rules cap training at *10 epochs*. Some runs in this paper do not comply: they exceed 10 epochs due to the project timeline and because parts of the work predated this year's rules. We report these results for analysis, not as an official challenge submission, while still respecting the 10M/100M word data limits.

[2] The single-GPU script has been removed in the meantime from the official release of GPT-BERT because it was not equivalent to the original multi-GPU training.

omit the $\geq 90\%$ change. MNTP masks a subset of input tokens and learns to predict each masked token using the hidden state at the preceding position.

Config.	BLiMP	BLiMP-Supp.
STRICT$_{\textit{rel. frequency}}$	0.653	0.637
STRICT$_{\textit{Zipf}}$	0.661	0.658
STRICT-SMALL$_{\textit{rel. frequency}}$	0.654	0.642
STRICT-SMALL$_{\textit{Zipf}}$	**0.663**	**0.661**

Table 2: Performance of the word-frequency scoring method on the STRICT and STRICT-SMALL corpora. The Zipf variant slightly outperforms relative frequency on both datasets.

3 Experiments

Unless noted, all scores in this section use the 2024 evaluator (Warstadt et al., 2023). The controlled-random baseline reaches **0.543** on BLiMP and **0.430** on BLiMP-Supplement, which serves as a true no-signal floor.

The results in Table 2 show that the word-frequency baseline is unexpectedly strong despite ignoring order and context. A Zipf weighting consistently outperforms relative frequencies, likely because the logarithmic scale better matches the log-likelihood scoring in the evaluator. Scores are essentially unchanged between STRICT and STRICT-SMALL, implying that corpus size contributes little once unigram statistics are learned.

N-gram language models (Table 3) are below the word-frequency baseline on both corpora. Accuracy increases with n and then plateaus, consistent with gains from local collocations rather than deeper structure. Notably, STRICT-SMALL often outperforms STRICT, suggesting that its distribution overlaps more with the linguistic patterns probed by the evaluation, despite its smaller size.

The LSTM models appear to be highly tokenizer-sensitive (Table 4). SuperBPE[3] shifts the distribution of units toward multi-word chunks, which helps slightly on dialogue/question phenomena but does not consistently improve core grammatical judgments. The results in Table 4 are directly comparable to the BabyHGRN (Haller et al., 2024) setup on STRICT-SMALL. BabyHGRN benchmarks sub-quadratic recurrent networks under the same BabyLM data budgets and includes an LSTM baseline. Our results show that tokenizer choice and vocabulary size affect accuracy on STRICT-

[3] https://huggingface.co/UW/OLMo2-8B-SuperBPE-t180k

KenLM	BLiMP	BLiMP-Supplement
2-gram$_{\textit{Strict}}$	0.596	0.552
2-gram$_{\textit{Strict-Small}}$	0.627	0.589
3-gram$_{\textit{Strict}}$	0.592	0.562
3-gram$_{\textit{Strict-Small}}$	0.632	0.587
4-gram$_{\textit{Strict}}$	0.598	0.572
4-gram$_{\textit{Strict-Small}}$	0.634	0.596
5-gram$_{\textit{Strict}}$	0.598	0.569
5-gram$_{\textit{Strict-Small}}$	0.634	0.603
6-gram$_{\textit{Strict}}$	0.598	0.570
6-gram$_{\textit{Strict-Small}}$	**0.633**	**0.606**

Table 3: Performance of KenLM n-gram models trained on the STRICT and STRICT-SMALL corpora, evaluated on BLiMP and BLiMP-Supplement. Despite the smaller data size, STRICT-SMALL often yields higher scores.

Vocab.	Tokenizer	BLiMP	BLiMP-Supp.
4k	SentencePiece BPE	0.644	**0.555**
	SentencePiece Unigram	0.646	0.547
	SuperBPE (trained)	**0.657**	0.536
8k	SentencePiece BPE	0.640	**0.581**
	SentencePiece Unigram	0.630	0.514
	SuperBPE (trained)	**0.661**	0.553
16k	SentencePiece BPE	0.607	0.522
	SentencePiece Unigram	**0.646**	0.537
	SuperBPE (trained)	0.613	**0.550**
	SuperBPE (pretrained)†	0.637	0.551

Table 4: LSTM performance on STRICT-SMALL grouped by tokenizer vocabulary size. The models are trained on the BabyLM dataset. Mid-size vocabularies (8k) yield the best BLiMP (SuperBPE–8k) and BLiMP-Supplement (BPE–8k), while Unigram is strongest at 4k; overall, tokenizer choice impacts accuracy more than vocabulary size. †Uses an externally pretrained vocabulary.

Dataset	Tokenizer	BLiMP	BLiMP-Supp.
Strict-Small	BPE	0.794	0.591
	Unigram	**0.796**	0.633
	SuperBPE	0.787	0.588
BabyCosmoFine	BPE	0.791	0.705
	Unigram	0.801	**0.715**
	SuperBPE	**0.803**	0.692

Table 5: 8k vocab GPT-BERT performance on STRICT-SMALL and BabyCosmoFine across tokenizers, evaluated on BLiMP and BLiMP-Supplement. Models trained on BabyCosmoFine score higher on BLiMP-Supplement, indicating better coverage of dialogue/question phenomena.

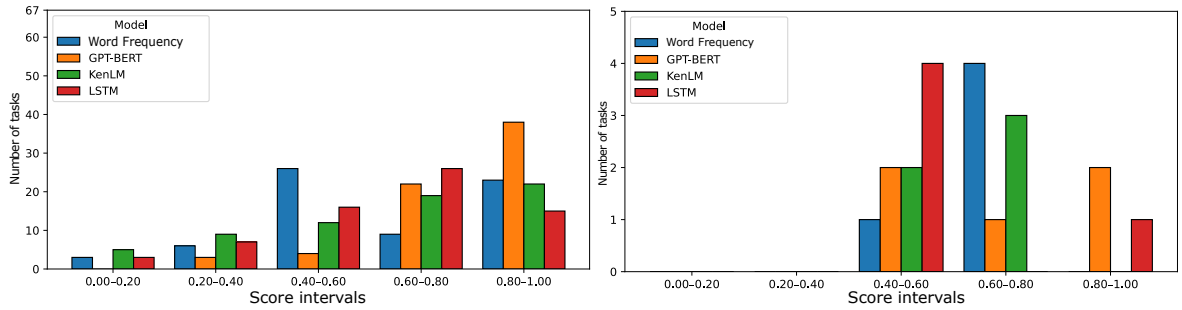

Figure 1: On the left, the distribution of model scores across BLiMP subtasks. On the right, the distribution of model scores across BLiMP-Supplement subtasks. GPT-BERT's scores cluster in the upper ranges, whereas the other models show a wider spread in performance. Only GPT-BERT and the LSTM achieve high scores for BLiMP-Supplement, showing how challenging these tasks can be.

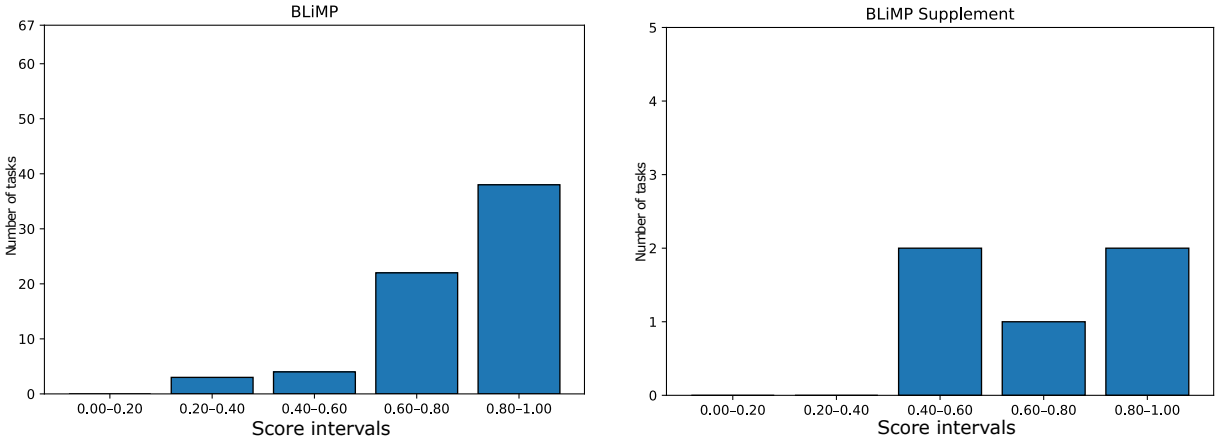

Figure 2: Distribution of GPT-BERT scores across BLiMP and BLiMP-Supplement subtasks. The share of BLiMP-Supplement subtasks with high scores is smaller than in BLiMP, indicating weaker performance on dialogue and question examples.

SMALL. The strongest BLiMP score is SuperBPE-8k, while BLiMP-Supplement is best with BPE-8k. Unigram is competitive at 4k but degrades at 8k. SuperBPE benefits from mid-size vocabularies on BLiMP and from larger vocabularies on BLiMP-Supplement.

Two consistent trends emerge: mid-size vocabularies favor dialogue/question phenomena across tokenizers, most clearly for BPE, while smaller vocabularies help Unigram on BLiMP. Overall, tokenizer family matters more than vocabulary size for LSTM models.

GPT-BERT (Charpentier and Samuel, 2024) is the strongest model in this study (see Table 5). Training it on BabyCosmoFine yields a clear lift on dialogue/question phenomena compared to STRICT-SMALL, pointing to a domain-coverage effect rather than purely scaling with data volume. Tokenizer choice matters less than for the LSTM; we treat GPT-BERT$_{(Unigram,\ BabyCosmoFine)}$ as the reference configuration.

4 Results Analysis

Figure 1 contains the distribution of BLiMP subtask accuracies for all systems. The results show that GPT-BERT clusters near the top, the LSTM sits in the mid–high range, and classical/lexical baselines spread widely. The pattern suggests that much of BLiMP can be addressed by stronger modeling capacity on top of lexical priors, with diminishing returns from short-context statistics alone. For the BLiMP-Supplement tasks, the distributions shift downward. Only GPT-BERT and the LSTM place substantial mass in the higher bins, underscoring the difficulty these tasks pose for most systems. This supports the view that BLiMP-Supplement probes conversational structures and question forms that benefit from models trained on dialogue-oriented corpora (e.g., BabyCosmoFine) and from tokenization schemes that stabilize sequence modeling. Figure 2 shows that GPT-BERT scores are concentrated in the upper bins for BLiMP, while BLiMP-Supplement is flat-

ter with fewer high-scoring subtasks. This gap mirrors our earlier results: core grammatical minimal pairs are largely handled, whereas dialog/question phenomena remain uneven, pointing to a domain-coverage effect rather than pure data scaling.

Table 6 shows GPT-BERT's strongest BLiMP subtasks: `irregular_past_participle_adjectives`, `determiner_noun_agreement_2`, and `determiner_noun_agreement_1`, all targeting morphology. Among the other systems, only the LSTM remains consistently competitive across these three; the word-frequency baseline does not stand out, and KenLM comes close on `determiner_noun_agreement_1`, consistent with short-range determiner–noun collocations. Representative items and each model's choice are given in Table 7.

Table 8 illustrates three BLiMP subtasks on which GPT-BERT performs worst. While the word-frequency baseline sometimes appears to answer these items correctly, this is largely an evaluation artifact. In 22 of the 67 BLiMP subtasks, the two sentences in each minimal pair are permutations of the same multiset of words. Any scorer that is invariant to word order assigns identical sentence scores to both sides of such pairs.

In the 2024 evaluator, ties are counted as correct, which inflates accuracy for permutation-equivalent items. When we exclude these 22 subtasks, the Zipf word-frequency mean drops from 0.663 to 0.498 on STRICT-SMALL, confirming that the initial score was driven by the artifact rather than genuine grammatical competence.

In the 2025 evaluator, identical sentence scores are still marked as correct by the evaluator. When the two candidates in a BLiMP minimal pair obtain the same sentence-level log-likelihood, the item is counted as correct. We construct a dummy model that assigns the same log-likelihood to both candidates in the benchmark. Concretely, at each scored position, we set the observed next-token logit to 0 and all other vocabulary logits to $-\infty$. The evaluator computes per-token log-probabilities with `log_softmax` and then sums only positions selected by the phrase mask. If a masked position leaves no valid entry (i.e., the row is all $-\infty$), `log_softmax` returns NaN. These NaNs propagate to both candidates' sentence totals, making them numerically indistinguishable; the evaluator treats this as a tie and marks the item correct. In practice, this yields a reported score of 100.0. For completeness, we also created a finite-negative variant that avoids NaNs by setting 0 on target tokens and -K on others, which yields a near-perfect score (99.69). Because non-target logits are finite, positions where the phrase mask removes the reference token contribute a constant offset $(-\log V)$ rather than 0. Minimal pairs can differ in how many such positions they contain, so the two sentence totals are not exactly equal; a small fraction of items cease to be ties, therefore the almost-perfect score.

5 Conclusions

We examine data-efficient pretraining in the BabyLM setting across classical and neural families while and varying tokenizer and corpus. Word–frequency signals already go far on BLiMP, obtaining scores as high as 0.66, exceeding LSTMs and n-gram language models. The frequency baseline achieves a 25% decrease in score apparent after removing the 22 permutation-based subtasks, where short-range collocations help. For n-gram language models STRICT-SMALL is a better choice than STRICT indicating that arbitrary changes in the dataset can have an impact of at least 0.05 in evaluation scores. This raises an important concern on when to decide if a system is actually stronger than another. The LSTM is sensitive to tokenization and GPT-BERT is the strongest model. Regarding corpus effects, we evaluated BabyCosmoFine only with GPT-BERT; in that setting, it yields clear improvements on dialogue and question-related phenomena compared to STRICT-SMALL.

Some BLiMP subtasks contain sentence pairs that are permutations of the same words. This led us to discover that, in both the 2024 and 2025 evaluators, when the two candidates receive the same sentence-level log-likelihood, the item is counted as correct; consequently, a no-signal system that enforces equal scores can report a score of 100.0.

Overall, tokenizer and data choices are relevant factors alongside model family at the 10M-word scale, and reporting across tokenizers helps make comparisons more informative.

Acknowledgments

This work was supported by the Romanian National Research Council (CNCS) through the Executive Agency for Higher Education, Research, Development and Innovation Funding (UEFISCDI) under grant PN-IV-P2-2.1-TE-2023-2007 InstRead.

BLiMP subtask	Description	GPT-BERT	WordFreq	KenLM	LSTM
irregular_past_participle_adjectives	Use of irregular past participles (e.g., *broken*, *hidden*) as adjectives.	0.996	0.612	0.622	0.874
determiner_noun_agreement_2	Number agreement between determiners and irregular nouns (e.g., *these geese* vs. *this geese*).	0.986	0.495	0.492	0.781
determiner_noun_agreement_1	Number agreement between determiners and regular nouns (e.g., *these dogs* vs. *this dogs*).	0.978	0.496	0.934	0.833
existential_there_quantifiers_2	Existential *there* with quantifiers and regular nouns (e.g., *there was every fish*).	0.236	1.000	0.667	0.418
left_branch_island_echo_question	Left-branch extraction constraint in echo questions (e.g., *Sara was insulting what student?*).	0.337	1.000	0.821	0.702
sentential_subject_island	Extraction from a sentential subject.	0.364	1.000	0.263	0.389

Table 6: GPT-BERT's best and worst BLiMP subtasks, compared with other systems.

BLiMP subtask	Sentences	GPT-BERT	WordFreq	KenLM	LSTM
irregular_past_participle_adjectives	Good: *The worn jacket was smooth.* Bad: *The wore jacket was smooth.*	Correct	Incorrect	Incorrect	Incorrect
determiner_noun_agreement_2	Good: *Robert hates that dancer.* Bad: *Robert hates those dancer.*	Correct	Correct	Correct	Correct
determiner_noun_agreement_1	Good: *Most waiters could break those couches.* Bad: *Most waiters could break those couch.*	Incorrect	Incorrect	Correct	Correct

Table 7: Examples from the three BLiMP subtasks on which GPT-BERT is strongest, showing each model's decision (Correct/Incorrect).

BLiMP subtask	Sentences	GPT-BERT	WordFreq	KenLM	LSTM
existential_there_quantifiers_2	Good: *All students weren't there noticing some box.* Bad: *There weren't all students noticing some box.*	Incorrect	Correct	Incorrect	Correct
left_branch_island_echo_question	Good: *Roger has noticed whose rivers?* Bad: *Whose has Roger noticed rivers?*	Incorrect	Correct	Correct	Correct
sentential_subject_island	Good: *Who would all cars' hurting Irene bore.* Bad: *Who would all cars' hurting bore Irene.*	Correct	Correct	Incorrect	Incorrect

Table 8: Examples from three BLiMP subtasks on which GPT-BERT is weakest, showing each model's decision (Correct/Incorrect). Although the word-frequency baseline sometimes appears correct on these items, we argue this reflects an evaluation artifact.

References

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. Smollm-corpus. https://huggingface.co/datasets/HuggingFaceTB/smollm-corpus. Accessed: 2025-06-06.

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. *Preprint*, arXiv:2502.10645.

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.

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

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 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):126.

Patrick Haller, Jonas Golde, and Alan Akbik. 2024. BabyHGRN: Exploring RNNs for sample-efficient language modeling. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural*

Language Learning, pages 82–94, Miami, FL, USA. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

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. *Preprint*, arXiv:2412.05149.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. 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. Extracting large parallel corpora from movie and tv subtitles. Technical Report 2016:22, University of Oslo.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, and Yejin Choi. 2025. Superbpe: Space travel for language models. *Preprint*, arXiv:2503.13423.

Brian MacWhinney. 2000. *The CHILDES Project: The Database*, 2 edition, volume 2 of *Tools for Analyzing Talk*. Psychology Press, Mahwah, NJ.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, and 1 others. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849.

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.

Robyn Speer. 2022. rspeer/wordfreq: v3.0 (v3.0.2).

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.

Walter JB Van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. 2014. Subtlex-uk: A new and improved word frequency database for british english. *Quarterly journal of experimental psychology*, 67(6):1176–1190.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

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

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