A Comparison of Elementary Baselines for BabyLM

Rareș Păpușoi, Sergiu Nisioi
Human Language Technologies Research Center
University of Bucharest
https://nlp.unibuc.ro
rarese19@yahoo.com, sergiu.nisioi@unibuc.ro

I. Introduction

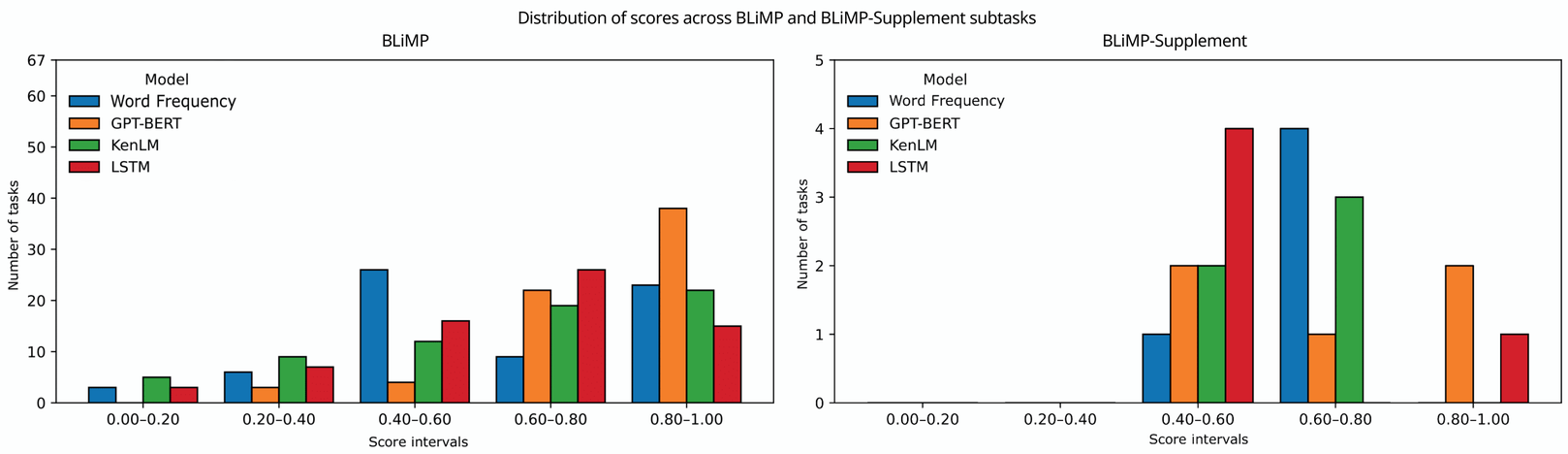
We present a comparison of elementary baselines for the BabyLM Challenge, focused on understanding what can be achieved with minimal modeling assumptions under strict data limits. We investigate how simple lexical predictors, classical probabilistic models, and neural architectures perform on the BabyLM evaluation suite. We consider three primary modelling approaches:

1. Trivial frequency-based baselines that ignore context and word order.
2. Classical n-gram and recurrent models.
3. A transformer model (GPT-BERT) adapted to the BabyLM constraints.

Core result: tokenization strongly affects accuracy, and a scoring artifact in permutation-equivalent pairs can inflate trivial baselines.

II. Results

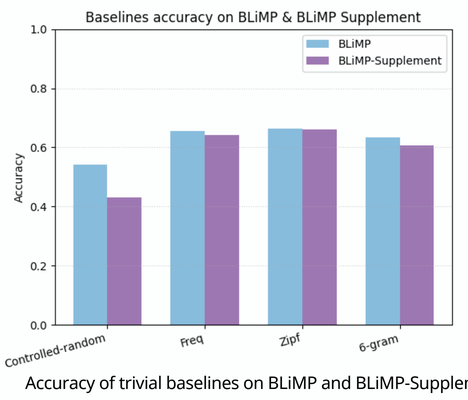
- Simple frequency-based baselines reach ~0.66 accuracy on BLiMP and BLiMP-Supplement, so strong scores do not automatically mean real grammatical generalization.
- Model performance is highly sensitive to tokenizer family and vocabulary size.
- BabyCosmoFine improves GPT-BERT on BLiMP-Supplement, suggesting that domain coverage matters more than just corpus size.
- Some BLiMP subtasks are permutation-equivalent, and ties are counted as correct; this can inflate reported accuracy for systems with no real signal.



Score distributions diverge by benchmark: on BLiMP, GPT-BERT concentrates in the higher range while other models cluster mid-range; on BLiMP-Supplement, scores shift lower for all models, with GPT-BERT retaining most of the remaining high-scoring tasks.

II. Baselines

- A word-frequency baseline and its Zipf-weighted variant reach ~0.66 accuracy on BLiMP.
- LSTM performance highly varies with tokenizer and vocabulary size.
- KenLM models performance increases with higher order.



Accuracy of trivial baselines on BLiMP and BLiMP-Supplement.

LSTM Scores with different tokenizers

Tokenizer	BLiMP	BLiMP Suppl.
SuperBPE (8k)	0.661	0.553
BPE (8k)	0.640	0.581
Unigram (4k)	0.646	0.547

Results show LSTM accuracy is tokenizer-dependent.

Sample from BLiMP

Correct Sentence

```
Roger has noticed whose rivers?
```

Incorrect Sentence

```
Whose has Roger noticed rivers?
```

BLiMP Sample

- A scorer that only sums word frequencies gives both sentences the same score on permutation-equivalent pairs.
- The BabyLM evaluation pipeline treats a tie as correct.
- Some subtasks inflate accuracy for some systems with no real grammatical signal.

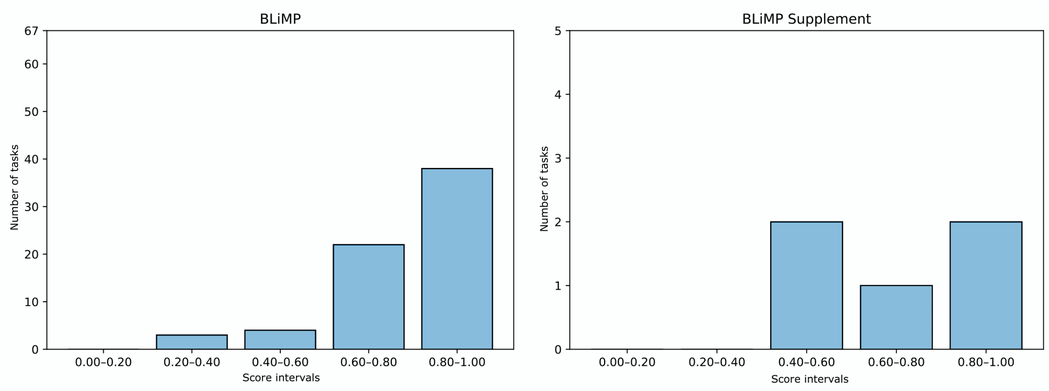
III. GPT-BERT

- GPT-BERT is the strongest model across all settings.
- Performance improves much more on BLiMP-Supplement than on BLiMP, meaning conversational / QA-style data helps specifically with dialogue phenomena.

GPT-BERT Scores

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

BabyCosmoFine improves GPT-BERT's performance on BLiMP-Supplement and yields the top BLiMP score, showing that corpus coverage and tokenizer choice matter.



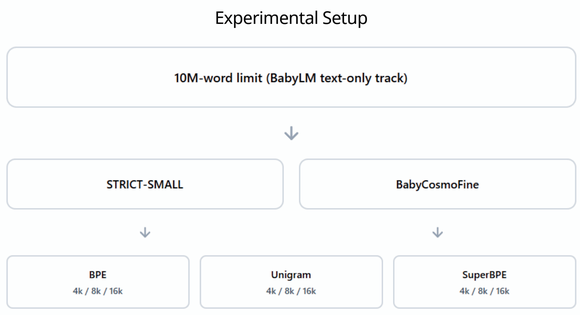
Distribution of GPT-BERT scores across BLiMP and BLiMP-Supplement subtasks.

The share of BLiMP-Supplement subtasks with high scores is smaller than in BLiMP, indicating weaker performance on dialogue and question examples.

- On BLiMP, most subtasks cluster in the high-accuracy range for GPT-BERT.
- On BLiMP-Supplement, the distribution is flatter and fewer subtasks reach high accuracy, which shows that dialogue / question-style phenomena are still harder.

IV. Data

- We work in the 10M-word text-only BabyLM setting.
- We compare STRICT-SMALL with BabyCosmoFine.
- BabyCosmoFine is a balanced mixture of BabyLM, FineWeb-Edu, and Cosmopedia, with equal contribution from each source.
- We train separate tokenizers (BPE, Unigram, SuperBPE) on these corpora.
- We study how corpus choice and tokenizer choice interact under the BabyLM constraints.



We train a separate tokenizer per corpus.

Tokenizer behaviour on the same sentence

BPE

```
By the way, I'm going to tell you something important!
```

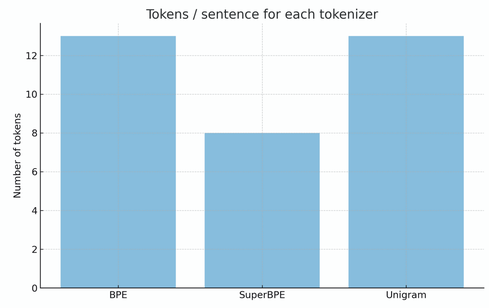
SuperBPE

```
By the way, I'm going to tell you something important!
```

Unigram

```
By the way, I'm going to tell you something important!
```

Example of tokenizer behaviour on the same sentence.



SuperBPE represents the same sentence with fewer tokens than BPE and Unigram, showing that tokenizer choice changes sequence length.

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.