# BabyLM's First Elementary Tasks: Character Tokenization Boosts Post-training for Counting and Spelling

**Bastian Bunzeck and Laurens Winkler and Sina Zarrieß**
Computational Linguistics, Department of Linguistics
Bielefeld University, Germany
{bastian.bunzeck, laurens.winkler, sina.zarriess}@uni-bielefeld.de

## Abstract

We expand the BabyLM paradigm by post-training small, English and German LMs (trained on only 10M words) with the simple, counting- and alphabet-based LMentry benchmark, for which we create a novel, German variant. We find that our BabyLMs outperform much larger, supposedly generalist models on this highly simplistic benchmark when replacing subword units with character tokenization. This gap can only be closed by further post-training generalist LMs on more LMentry-style data. Our findings underline how even a small amount of language-only pre-training data in combination with a suitable tokenization scheme can provide useful inductive biases for task-specific post-training.

## 1 Introduction

In the last three years, the BabyLM challenge (Warstadt et al., 2023; Hu et al., 2024; Charpentier et al., 2025) has developed into a research paradigm that investigates small LMs trained in data-scarce settings (only 10M/100M words), enabling researchers to identify which architectural decisions and types of training data support specific capabilities. To date, this paradigm has mostly been restricted to genuine *language* modeling; approaches like post-training or evaluation on non-linguistic, knowledge-heavy tasks have been regarded largely out of scope. However, a major source of fascination that surrounds LLMs stems from the fact that these models develop aspects of knowledge and reasoning that exceed the text generation capabilities one would intuitively expect from their training objectives. Unfortunately, for truly *large* LLMs it is difficult to study how these abilities emerge, since they are trained on enormous datasets, which complicates precise ablations and interventions.

We extend the BabyLM scope to rudimentary tasks and show that such small models are not only useful to study language *per se*, but can also shed light on the emergence of simplistic knowledge and basic reasoning abilities. We do so by investigating the ability of BabyLMs to learn elementary counting/spelling tasks from the English LMentry benchmark (Efrat et al., 2023) and a newly created German variant. In stark contrast to **L**LMs, we pre-train on an extremely small corpus of 10M words and then go beyond the established BabyLM paradigm by post-training our models on the LMentry benchmark. As tokenization has been shown to both influence performance on mathematical tasks (Golkar et al., 2023; McCoy et al., 2024) and BabyLM benchmarks (Goriely et al., 2024; Bunzeck et al., 2025a), we compare subword and character-based models.

We find that post-trained character BabyLMs achieve impressive performance on LMentry, outperforming comparable subword LMs at almost all tasks. This profound performance gap also persists when comparing our BabyLMs against small off-the-shelf LLMs (SmolLM2 and Llama-3.2). Although trained on trillions of tokens and post-trained across a wide variety of tasks, they underperform our task-specific models drastically. Only when additionally trained on LMentry do they reach similar benchmark scores to our BabyLMs. This shows how powerful BabyLMs can be when tokenization and task-specific post-training are matched carefully, and also calls for a reconsideration of fine-tuning often being considered unnecessary in the age of off-the-shelf models, while aligning with growing interest in subword-less language models (Pagnoni et al., 2025; Minixhofer et al., 2025).

## 2 Related work

The term *BabyLM* is commonly used for LMs trained on very little, developmentally plausible data (Huebner et al., 2021; Warstadt et al., 2023). The major, self-imposed limitation in training BabyLMs therefore differs from other small LMs,

e.g. SmolLM, which are *small* in parameter size, but trained on enormous amounts of curated data (Allal et al., 2025). Standard evaluation tasks for such BabyLMs are not downstream-oriented, but rather linguistic/cognitive in nature. As of 2025, the evaluation pipeline (Charpentier et al., 2025) consists of zero-shot minimal pairs tasks for syntax, semantics and world knowledge (e.g., BLiMP, Warstadt et al., 2020; COMPS, Misra et al., 2023; EWoK, Ivanova et al., 2024). In further tasks, model probabilities are correlated with cognitive measures like word-level age of acquisition (Chang and Bergen, 2022). The only downstream evaluation involves classical fine-tuning on (Super)GLUE (Wang et al., 2018, 2019). This diverges from current standard practices and evaluations in language modeling, for example instruction tuning (Wang et al., 2022) and post-training (Kumar et al., 2025) on tasks formulated in natural language.

One target domain for this kind of post-training are elementary counting and spelling abilities, as these remain a challenge for small LMs. A simple benchmark is LMentry, which tests for basic word-level knowledge and elementary reasoning abilities beyond language (like counting, finding first/last letters/words, etc.). Efrat et al. (2023) report that especially base models, but also large instruction-tuned models (175B parameters) frequently fail such tasks, which are easily solved by adult humans. Due to this basic task nature, we identify LMentry as a highly suitable challenge for BabyLMs that bridges the gap between the aforementioned linguistic tests and downstream tasks. Interestingly, a frequently identified problem in mathematical and character-based evaluations is tokenization: scientific reasoning (Golkar et al., 2023), LaTeX and STEM tasks (Altıntaş et al., 2025), or the correct disambiguation of existing and non-words (Goriely et al., 2024; Bunzeck et al., 2025a) benefit tremendously from replacing subwords with open-vocabulary language modeling.

Concurrent to our work, Moroni et al. (2025) present Multi-LMentry, an extension of LMentry for nine typologically diverse languages (including German), and confirm that state-of-the-art multilingual LLMs also fall short of human performance (compared to L2 speakers of Basque), especially for non-English data. However, in contrast to our approach, they only evaluate models of different sizes in a zero-shot setting, and do not additionally fine-tune any models on their custom LMentry datasets.

## 3 Methods

**Pre- and post-training** We train small Llama models on 10M tokens of English (Charpentier et al., 2025) or German (Bunzeck et al., 2025b) data with the `transformers` library (Wolf et al., 2020). For each language, we pre-train a BPE model and a character model (see Appendix C for a comparison of tokenizer quality). We then post-train the models on 80% of the English LMentry data (700k words) or the novel, German LMentry dataset (500k words). As is common in instruction-tuning, we convert LMentry to question-answer pairs and further train the base models with next-token-prediction on these. The held-out 20% are used for evaluation. We post-train both English and German models for each singular task individually, and two additional models on all tasks. Full details are presented in Appendix A.

**German LMentry dataset creation** We create a German variant of LMentry (Efrat et al., 2023) by combining templates for question types with different words/numbers, resulting in questions of the following kind: *Von den Zahlen 510 und 610, welche ist größer?* (*Of the numbers 510 and 610, which is larger?*, cf. Appendix E for an overview of all tasks and possible answers). Word lists for the tasks were generated with ChatGPT-4o, independently of the English data. While we used no formal validation for our dataset, the items were checked by the authors, who are all native speakers of German. For tasks that require additional linguistic information (e.g. `rhyming word`) we do not create German variants due to a lack of readily available rhyming dictionaries and similarity measures. All in all, our dataset contains 19 tasks with 3000 examples each (like English LMentry). We share our dataset on HuggingFace.

**Evaluation** We evaluate our models in the LMentry question-answer setup on the held-out 20% of the dataset. We use `outlines` (Willard and Louf, 2023) to constrain text generation to answers allowed for the respective task (cf. Appendix E for possible completions and D for free-form generation). To assess our models on a more standard BabyLM benchmark, we also evaluate them on MultiBLiMP (Jumelet et al., 2026b), which contains linguistic minimal pairs with matched grammatical and ungrammatical sentences. We report accuracies for correct preferences across the whole English and German datasets. Further, we compare

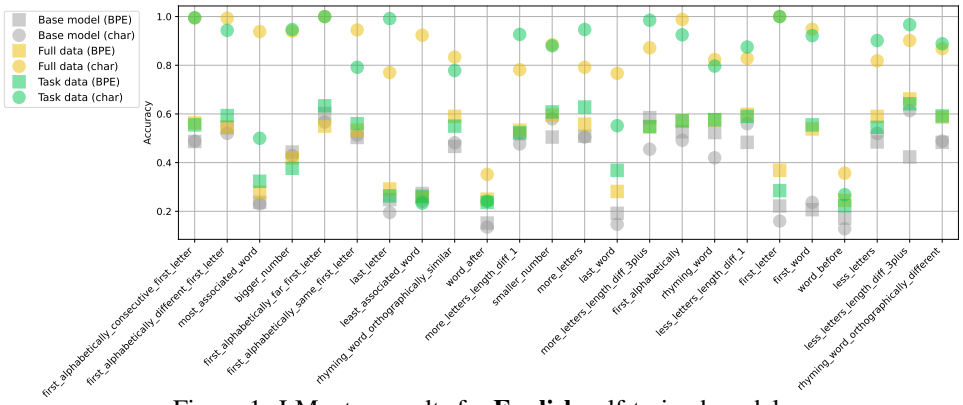


Figure 1: LMentry results for **English** self-trained models.

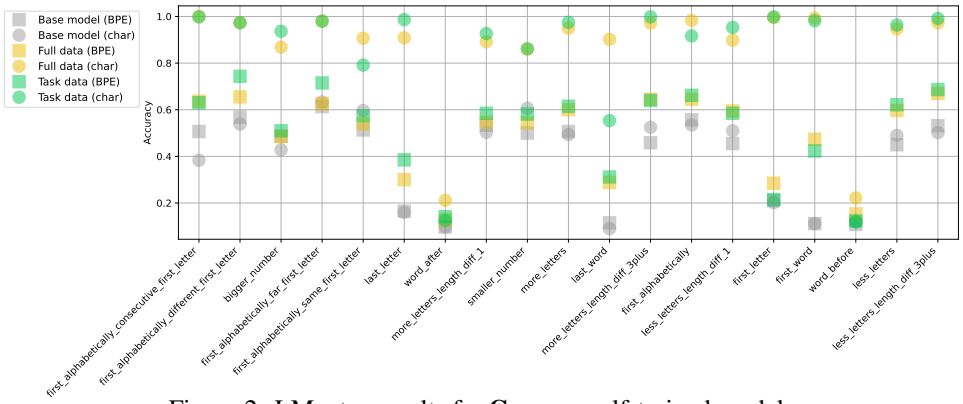


Figure 2: LMentry results for **German** self-trained models.

the performance of our models with the considerably larger base and instruction-tuned versions of the multilingual Llama-3.2-1B model (Meta AI, 2024). As ablations, we additionally also post-train two models from the SmolLM2 family (Allal et al., 2025) on LMentry, in the same manner as our own models. We focus on the 135M and 360M variants, as the former is (roughly) parameter-matched to our BabyLMs, while fine-tuning the latter is still not overly resource-demanding. Finally, Appendix F presents further results for base and instruction-tuned variants of SmolLM2 and Llama-3.2-3B.

## 4 Results

**General results** We present the results for all tasks visually in Figures 1 and 2, and numerically in Table 1. (While we did not run multiple seeds, a binomial 95% CI with n=600 held-out items per task roughly corresponds to ±4%. Therefore, differences under approx. 5% are not interpreted).

Unsurprisingly, our non-post-trained base models (gray markers) do not perform exceptionally well on LMentry, very rarely surpassing an accuracy of 50% for specific tasks. In general, fine-tuning with LMentry data then consistently improves performance for BPE and character models, both for the English and the German data. However, the magnitude of the improvement due to fine-tuning differs drastically between BPE and character models: whereas the former exhibit modest improvements (between 2-10%), the improvements for the character models are much higher, sometimes reaching close to perfect accuracy. As the base models (pre-trained only on 10M words of English/German BabyLM data) feature only very small performance differences between BPE and character tokenization, this enormous discrepancy in improvement suggests that the character models do not learn the abilities tested by LMentry (and exemplified through only 2400 examples per task) during pre-training, but pick them up much better than the BPE models in our post-training regimen.

**Individual tasks** With regard to the individual tasks it can be said that overall trends are generally stable across base/post-trained models and the different tokenization schemes. Challenging tasks (like `word after`, `word before` or `last word`) exhibit similarly bad performance across all model types and for both languages. In contrast, performance on the remaining tasks varies between almost perfect scores (80-100%) for many of the post-trained characters models and somewhat worse (60-80%) scores for the BPE models.

| Task | English | | | | | | German | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Char | | BPE | | Llama-3.2-1B | | Char | | BPE | | Llama-3.2-1B | |
| | Single | Full | Single | Full | Base | Instruct | Single | Full | Single | Full | Base | Instruct |
| smaller number | 0.88 | **0.89** | 0.60 | 0.60 | 0.47 | 0.57 | **0.86** | **0.86** | 0.58 | 0.55 | 0.40 | 0.61 |
| bigger number | **0.95** | 0.94 | 0.38 | 0.24 | 0.61 | 0.44 | **0.94** | 0.87 | 0.51 | 0.49 | 0.52 | 0.38 |
| last letter | **0.99** | 0.77 | 0.26 | 0.29 | 0.27 | 0.31 | **0.99** | 0.91 | 0.39 | 0.30 | 0.27 | 0.22 |
| first letter | **1.00** | **1.00** | 0.29 | 0.39 | 0.22 | 0.29 | **1.00** | **1.00** | 0.22 | 0.29 | 0.21 | 0.14 |
| more letters | **0.95** | 0.79 | 0.63 | 0.56 | 0.51 | 0.58 | **0.98** | 0.95 | 0.61 | 0.60 | 0.49 | 0.53 |
| more letters length diff 1 | **0.93** | 0.78 | 0.52 | 0.53 | 0.49 | 0.54 | **0.93** | 0.89 | 0.59 | 0.55 | 0.47 | 0.48 |
| more letters length diff 3plus | **0.99** | 0.87 | 0.55 | 0.55 | 0.52 | 0.65 | **1.00** | 0.97 | 0.64 | 0.65 | 0.46 | 0.57 |
| less letters | **0.90** | 0.82 | 0.55 | 0.59 | 0.50 | 0.41 | **0.96** | 0.95 | 0.62 | 0.60 | 0.54 | 0.48 |
| less letters length diff 1 | **0.88** | 0.83 | 0.59 | 0.60 | 0.48 | 0.42 | **0.95** | 0.90 | 0.59 | 0.60 | 0.51 | 0.47 |
| less letters length diff 3plus | **0.97** | 0.90 | 0.64 | 0.66 | 0.49 | 0.44 | **0.99** | 0.97 | 0.69 | 0.67 | 0.53 | 0.42 |
| word before | 0.27 | **0.36** | 0.22 | 0.25 | 0.16 | 0.21 | 0.12 | **0.22** | 0.12 | 0.15 | 0.12 | 0.11 |
| word after | 0.24 | **0.35** | 0.24 | 0.25 | 0.17 | 0.21 | 0.16 | **0.21** | 0.14 | 0.13 | 0.08 | 0.11 |
| first word | 0.92 | **0.95** | 0.56 | 0.54 | 0.19 | 0.19 | 0.98 | **0.99** | 0.42 | 0.47 | 0.18 | 0.09 |
| last word | 0.55 | **0.77** | 0.37 | 0.28 | 0.19 | 0.24 | 0.55 | **0.90** | 0.31 | 0.29 | 0.10 | 0.08 |
| first alphabetically | 0.93 | **0.99** | 0.57 | 0.57 | 0.39 | 0.54 | 0.92 | **0.98** | 0.66 | 0.65 | 0.49 | 0.44 |
| first alphabetically same first | 0.80 | **0.95** | 0.56 | 0.53 | 0.49 | 0.46 | 0.79 | **0.91** | 0.57 | 0.54 | 0.49 | 0.44 |
| first alphabetically cons. first | **1.00** | **1.00** | 0.60 | 0.56 | 0.52 | 0.52 | **1.00** | **1.00** | 0.63 | 0.64 | 0.64 | 0.51 |
| first alphabetically far first | **1.00** | **1.00** | 0.63 | 0.55 | 0.38 | 0.57 | **0.98** | **0.98** | 0.71 | 0.63 | 0.51 | 0.54 |
| first alphabetically diff. first | 0.94 | **0.99** | 0.59 | 0.55 | 0.46 | 0.49 | **0.97** | **0.97** | 0.74 | 0.66 | 0.38 | 0.59 |
| rhyming word | 0.80 | **0.82** | 0.58 | 0.58 | 0.51 | 0.51 | - | - | - | - | - | - |
| rhyming word orthographically diff. | **0.89** | 0.87 | 0.59 | 0.59 | 0.51 | 0.51 | - | - | - | - | - | - |
| rhyming word orthographically sim. | 0.78 | **0.83** | 0.55 | 0.59 | 0.51 | 0.51 | - | - | - | - | - | - |
| least associated word | 0.23 | **0.92** | 0.26 | 0.26 | 0.22 | 0.22 | - | - | - | - | - | - |
| most associated word | 0.50 | **0.94** | 0.32 | 0.29 | 0.22 | 0.22 | - | - | - | - | - | - |
| MultiBLiMP | - | 0.74 | - | 0.80 | **0.99** | 0.98 | - | 0.79 | - | 0.82 | 0.93 | 0.88 |

Table 1: Full accuracies for all tasks (best performing models marked in **boldface** type), "Char" and "BPE" refer to our post-trained BabyLMs.

**Single- vs. full-task tuning** Table 1 shows that especially the character models fine-tuned on the full LMentry data often underperform those trained on only a single task. Many of the tasks where the full fine-tune is superior to the task-specific fine-tune are those that feature a larger number of possible answers (cf. Appendix E) and are therefore more complex. It is plausible that these tasks therefore benefit from the models being exposed to a wider variety and higher number of similar/related, yet different tasks. For the BPE models, these patterns are less pronounced or stable, which also aligns with their general performance deficits.

**Task-specific vs. off-the-shelf models** In comparison to the explicitly not LMentry-tuned Llama-3.2-1B model, we find that our small character LMs outperform it on LMentry tasks in English and German, despite Llama featuring over eight times more parameters than our BabyLMs and being trained on considerably more data (cf. Appendix B). The general instruction-tuning applied for Llama-3.2 has no straightforward effect on LMentry performance and does not lead to overall improvement. Interestingly, the larger Llama models reach almost perfect accuracies on MultiBLiMP, which shows that their linguistic capability is much higher than that of our BabyLMs (which perform well above the chance baseline, but fall 10–25% short of the Llama models). These results also hold for even larger models (1.7B/3B parameters, Appendix F).

**Fine-tuning of non-BabyLMs** Table 2 reports results for two models from the SmolLM2 family (Allal et al., 2025), SmolLM2-135M and SmolLM2-360M, which were post-trained on the (full) English/German LMentry datasets in the same manner as our BabyLMs, both in their base and already instruction-tuned versions. Importantly, the SmolLM models are pre-trained on English data only, so the German tasks require additional cross-lingual transfer and adaptation to the novel, but closely related language.

In general, the LMentry-tuned non-BabyLMs perform very well across all tasks, reaching full accuracy on many of them. Our LMentry-tuned BabyLMs only beat them on a very small selection of tasks: for the alphabetical-ordering family, first-letter/first-word and German MultiBLiMP, our models outperform the size-matched SmolLM2-135M. Against the 360M models, only three winning tasks survive, all in German: `first alphabetically` (0.98 vs 0.95), `same first` (0.91 vs 0.79) and `cons. first` (1.00 vs 0.93), as well as MultiBLiMP (0.82 vs 0.64, but unsurprising as the SmolLM training data is English-only). For the same-size comparison, this means that our BabyLMs only win on tasks that depend solely on character-level ordering and no additional world knowledge. These advantages further diminish with more parameters and only remain for German tasks which require cross-lingual transfer and are therefore even more challenging.

| Task | English | | | | German | | | |
|---|---|---|---|---|---|---|---|---|
| | SmolLM2-135M | | SmolLM2-360M | | SmolLM2-135M | | SmolLM2-360M | |
| | Base | Instruct | Base | Instruct | Base | Instruct | Base | Instruct |
| smaller number | 0.98 | 0.99 | **1.00** | **1.00** | 0.98 | 0.99 | **1.00** | **1.00** |
| bigger number | 0.98 | **1.00** | **1.00** | **1.00** | 0.98 | **1.00** | **1.00** | **1.00** |
| last letter | 0.93 | 0.96 | 0.98 | **1.00** | 0.94 | 0.95 | 0.98 | **0.99** |
| first letter | 0.99 | **1.00** | **1.00** | **1.00** | 0.77 | **0.99** | 0.98 | **0.99** |
| more letters | 0.99 | 0.99 | **1.00** | **1.00** | 0.82 | 0.99 | **1.00** | **1.00** |
| more letters length diff 1 | 0.99 | **1.00** | 0.99 | **1.00** | 0.81 | 0.87 | **0.97** | 0.96 |
| more letters length diff 3plus | 0.99 | **1.00** | **1.00** | **1.00** | 0.83 | 0.99 | **1.00** | **1.00** |
| less letters | 0.99 | 0.99 | **1.00** | **1.00** | 0.91 | 0.99 | **1.00** | **1.00** |
| less letters length diff 1 | 0.97 | 0.98 | 0.99 | **1.00** | 0.83 | 0.91 | **0.98** | 0.96 |
| less letters length diff 3plus | **1.00** | **1.00** | **1.00** | **1.00** | 0.95 | 0.99 | **1.00** | **1.00** |
| word before | 0.85 | 0.88 | 0.91 | **0.94** | 0.59 | 0.77 | 0.84 | **0.85** |
| word after | 0.96 | 0.95 | 0.97 | **0.99** | 0.84 | 0.91 | 0.97 | **0.98** |
| first word | 0.80 | **0.92** | 0.91 | 0.85 | 0.82 | **0.99** | 0.92 | 0.95 |
| last word | 0.97 | **0.99** | 0.98 | **0.99** | 0.84 | 0.91 | **0.94** | 0.93 |
| first alphabetically | 0.96 | 0.94 | **0.99** | **0.99** | 0.86 | 0.85 | 0.92 | **0.95** |
| first alphabetically same first | 0.91 | 0.91 | **0.95** | 0.94 | 0.67 | 0.71 | 0.78 | **0.79** |
| first alphabetically cons. first | 0.92 | 0.92 | 0.96 | **0.97** | 0.84 | 0.84 | **0.93** | 0.92 |
| first alphabetically far first | **1.00** | 0.99 | **1.00** | **1.00** | 0.86 | 0.87 | **0.98** | **0.98** |
| first alphabetically diff. first | 0.97 | 0.96 | **0.99** | **0.99** | 0.90 | 0.85 | **0.99** | 0.97 |
| rhyming word | 0.93 | 0.95 | **0.98** | 0.97 | - | - | - | - |
| rhyming word ortho. different | 0.93 | 0.95 | **0.98** | 0.96 | - | - | - | - |
| rhyming word ortho. similar | 0.94 | 0.95 | **0.99** | 0.98 | - | - | - | - |
| least associated word | **1.00** | **1.00** | **1.00** | **1.00** | - | - | - | - |
| most associated word | **1.00** | **1.00** | **1.00** | **1.00** | - | - | - | - |
| MultiBLiMP | 0.90 | 0.89 | **0.93** | 0.91 | **0.65** | 0.64 | 0.64 | 0.64 |

Table 2: Full accuracies for SmolLM2 finetuned models on the English and German LMentry benchmarks, best results per language marked in **boldface**. Tasks unavailable for German are marked with -.

## 5 Discussion and conclusion

Our study is among the first to show that the capabilities of BabyLMs can extend way beyond "core" language learning when post-trained with narrow, task-related data. This aligns with a concurrent push for data-efficient post-training (cf. Luo et al., 2025; Allal et al., 2025) and the comparability of BabyLMs with larger LMs (e.g., Trhlík et al., 2026). Additionally, it opens promising avenues for future research, like a) investigating how pre-training and post-training interact in extreme low-data regimes, i.e. what must be learned during pre-training for post-training to succeed at all, or b) looking at what structures (missed by current evaluations) character-level models acquire that enable their post-training efficiency.

Additionally, our results add to concurrent findings on the effectiveness of subword-free tokenization (Pagnoni et al., 2025; Moryossef et al., 2025; Minixhofer et al., 2025), which aids tasks that require precise character-level understanding. The much stronger performance of character-based models on LMentry suggests that working at a finer granularity of input offers advantages in data-scarce settings, potentially due to the models directly modeling the structures that govern individual symbols and their combinations. This confirms previous findings on math-based tasks (Golkar et al., 2023; Altıntaş et al., 2025), where appropriate tokenization is equally important as in BabyLMs that model linguistic processes below the syntactic level (Goriely et al., 2024; Goriely and Buttery, 2025; Bunzeck and Zarrieß, 2025; Fusco et al., 2025).

A final, equally important implication of our findings is that the scaling paradigm or "bitter lesson" in machine learning should be reconsidered in light of task-specific demands. The idea that bigger models and instruction-tuning on more complex tasks always lead to better performance is challenged by our results, especially when looking at the off-the-shelf models we compare our results with. Even post-training on enormous, highly varied instruction-tuning datasets featuring many different tasks does not help LLMs to perform these tasks. This finding is in line with the concurrent Multi-LMentry (Moroni et al., 2025), which also reports low accuracies for off-the-shelf models. In contrast, a task-fitting architecture with a little task-specific post-training pushes accuracies on many tasks to almost optimal performance. While the same tends to hold for larger models, as our ablations show, we argue that small models may offer a more efficient and computationally affordable path to achieving specific types of reasoning when optimized and post-trained correctly, and could, for example, be deployed on-device instead of relying on the availability of hosted models.

## Limitations

Currently, our experiments are limited by only testing one model size for our self-trained models. It would certainly be interesting to test if models can be downscaled even further, both parameter-wise and with regard to pre-training data, while still retaining this impressive performance. Further future work could also expand upon our findings by investigating the connection between pre- and post-training in more detail. It still remains opaque how both types of training push each other – does post-training performance depend crucially on the pre-training data? Is pre-training performance already saturated with BabyLM data or could this type of developmentally plausible input be enhanced with small amounts of synthetic or non-linguistic data? How far do post-trained models generalize to novel task types in LMentry style that they were not explicitly trained on? And what is the overall capacity of BabyLMs – can they learn an infinite amount of tasks via post-training, or are they saturated early?

A further limitation is the selection of languages that we train our models on. For example, the study of typologically more diverse languages would be a worthwhile goal. While most of the tasks in LMentry, e.g. those concerned with letter counting, are not directly translatable to logographic writing systems (like Chinese or Japanese) or even abjad scripts (like Hebrew or Arabic), an extension to other Indo-European languages is feasible (and indeed possible, cf. Multi-LMentry by Moroni et al., 2025, which was not yet released publicly when we first created our German LMentry variant) and would aid BabyLM evaluation, especially in light of the release of novel, linguistically-oriented datasets like BabyBabelLM (Jumelet et al., 2026a).

Finally, it should also be noted that character-based language modeling is not the only form of vocabulary-free language modeling (Mielke et al., 2021). Especially byte-level language models have recently begun to receive increased attention (Pagnoni et al., 2025; Moryossef et al., 2025), and examples like Bolmo show that subword-based models can be adjusted to the byte level with only little additional training (Minixhofer et al., 2025). It remains open to further inquiry if these models are able to better pick up on the type of fine-grained character-level knowledge needed for LMentry tasks.

## References


Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. 2025. SmolLM2: When Smol Goes Big – Data-Centric Training of a Small Language Model. *Preprint*, arXiv:2502.02737.

Gül Sena Altıntaş, Malikeh Ehghaghi, Brian Lester, Fengyuan Liu, Wanru Zhao, Marco Ciccone, and Colin Raffel. 2025. TokSuite: Measuring the Impact of Tokenizer Choice on Language Model Behavior. *arXiv preprint*.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2025a. Small language models also work with small vocabularies: Probing the linguistic abilities of grapheme- and phoneme-based baby llamas. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6039–6048, Abu Dhabi, UAE. Association for Computational Linguistics.

Bastian Bunzeck, Daniel Duran, and Sina Zarrieß. 2025b. Do construction distributions shape formal language learning in German BabyLMs? In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pages 169–186, Vienna, Austria. Association for Computational Linguistics.

Bastian Bunzeck and Sina Zarrieß. 2025. Subword models struggle with word learning, but surprisal hides it. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 286–300, Vienna, Austria. Association for Computational Linguistics.

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

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Y. Hu, Jing Liu, Jaap Jumelet, Tal Linzen, Aaron Mueller, Candance Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Gotlieb Wilcox, and Adina Williams. 2025. Findings of the third BabyLM challenge: Accelerating language modeling research with cognitively plausible data. In *Proceedings of the First BabyLM Workshop*, pages 399–420, Suzhou, China. Association for Computational Linguistics.

Avia Efrat, Or Honovich, and Omer Levy. 2023. LMentry: A Language Model Benchmark of Elementary Language Tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10476–10501, Toronto, Canada. Association for Computational Linguistics.

Yihao Feng, Shelby Heinecke, Thai Hoang, Shirley Kokane, Tian Lan, Zhiwei Liu, Zuxin Liu, Rithesh Murthy, Juan Niebles, Silvio Savarese, Juntao Tan, Huan Wang, Caiming Xiong, Liangwei Yang, Weiran Yao, Jianguo Zhang, and Ming Zhu. 2024. APIGen: Automated PIpeline for Generating Verifiable and Diverse Function-Calling Datasets. In *Advances in Neural Information Processing Systems 37*, pages 54463–54482, Vancouver, BC, Canada. Neural Information Processing Systems Foundation, Inc. (NeurIPS).

Achille Fusco, Maria Letizia Piccini Bianchessi, Tommaso Sgrizzi, Asya Zanollo, and Cristiano Chesi. 2025. Linguistic Units as Tokens: Intrinsic and Extrinsic Evaluation with BabyLM. In *Proceedings of the First BabyLM Workshop*, pages 496–507, Suzhou, China. Association for Computational Linguistics.

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

Siavash Golkar, Mariel Pettee, Michael Eickenberg, Alberto Bietti, Miles Cranmer, Geraud Krawezik, Francois Lanusse, Michael McCabe, Ruben Ohana, Liam Parker, Bruno Régaldo-Saint Blancard, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. 2023. xVal: A Continuous Numerical Tokenization for Scientific Language Models. *arXiv preprint*.

Zébulon Goriely and Paula Buttery. 2025. BabyLM's First Words: Word Segmentation as a Phonological Probing Task. *arXiv preprint*.

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.

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.

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

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2024. Elements of World Knowledge (EWOK): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint*.

Jaap Jumelet, Abdellah Fourtassi, Akari Haga, Bastian Bunzeck, Bhargav Shandilya, Diana Galvan-Sosa, Faiz Ghifari Haznitrama, Francesca Padovani, Francois Meyer, Hai Hu, Julen Etxaniz, Laurent Prevot, Linyang He, María Grandury, Mila Marcheva, Negar Foroutan, Nikitas Theodoropoulos, Pouya Sadeghi, Siyuan Song, Suchir Salhan, Susana Zhou, Yurii Paniv, Ziyin Zhang, Arianna Bisazza, Alex Warstadt, and Leshem Choshen. 2026a. BabyBabelLM: A multilingual benchmark of developmentally plausible training data. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3297–3329, Rabat, Morocco. Association for Computational Linguistics.

Jaap Jumelet, Leonie Weissweiler, Joakim Nivre, and Arianna Bisazza. 2026b. MultiBLiMP 1.0: A Massively Multilingual Benchmark of Linguistic Minimal Pairs. *Transactions of the Association for Computational Linguistics*, 14:193–216.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip H. S. Torr, Fahad Shahbaz Khan, and Salman Khan. 2025. LLM Post-Training: A Deep Dive into Reasoning Large Language Models. *arXiv preprint*.

Junyu Luo, Bohan Wu, Xiao Luo, Zhiping Xiao, Yiqiao Jin, Rong-Cheng Tu, Nan Yin, Yifan Wang, Jingyang Yuan, Wei Ju, and Ming Zhang. 2025. A Survey on Efficient Large Language Model Training: From Data-centric Perspectives. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30904–30920, Vienna, Austria. Association for Computational Linguistics.

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. 2024. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121.

Meta AI. 2024. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models.

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. 2021. Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP. *arXiv preprint*.

Benjamin Minixhofer, Tyler Murray, Tomasz Limisiewicz, Anna Korhonen, Luke Zettlemoyer, Noah A. Smith, Edoardo M. Ponti, Luca Soldaini, and Valentin Hofmann. 2025. Bolmo: Byteifying the Next Generation of Language Models. *arXiv preprint*.

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.

Luca Moroni, Javier Aula-Blasco, Simone Conia, Irene Baucells, Naiara Perez, Silvia Paniagua Suárez, Anna Sallés, Malte Ostendorff, Júlia Falcão, Guijin Son, Aitor Gonzalez-Agirre, Roberto Navigli, and Marta Villegas. 2025. Multi-LMentry: Can Multilingual LLMs Solve Elementary Tasks Across Languages? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34114–34145, Suzhou, China. Association for Computational Linguistics.

Amit Moryossef, Clara Meister, Pavel Stepachev, and Desmond Elliott. 2025. Back to Bytes: Revisiting Tokenization Through UTF-8. *Preprint*, arXiv:2510.16987.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. 2025. Byte Latent Transformer: Patches Scale Better Than Tokens. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258, Vienna, Austria. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. FineWeb2: One Pipeline to Scale Them All – Adapting Pre-Training Data Processing to Every Language. *arXiv preprint*.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

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

Filip Trhlík, Andrew Caines, and Paula Buttery. 2026. Bias Dynamics in BabyLMs: Towards a Compute-Efficient Sandbox for Democratising Pre-Training Debiasing. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 33356–33377, San Diego, California, United States. Association for Computational Linguistics.

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

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

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Gotlieb Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–6, Singapore. Association for Computational Linguistics.

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

Brandon T. Willard and Rémi Louf. 2023. Efficient Guided Generation for Large Language Models. *Preprint*, arXiv:2307.09702.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz,

Joe Davison, Sam Shleifer, Patrick Von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

## A Pre- and post-training details

All our models share the following hyperparameters: a hidden size of 768, an intermediate size of 3072, 12 hidden layers and 12 attention heads. The subword models feature a BPE vocabulary of size 16,000 learned over the complete training data, resulting in 137,841,408 parameters. The character models feature a vocabulary of size 76, resulting in 113,382,144 parameters. As for tokenization, we opted for the `LlamaTokenizerFast` class from `transformers`, as it yielded the best performance for our character-level models and we noticed some performance drops when training with the comparable `AutoTokenizerFast` and `GPT2TokenizerFast`.

Our complete training run was conducted on a NVIDIA GeForce RTX 4070 SUPER. We pre-train all models for five epochs with a learning rate of 3e-4 on 10M lexical tokens of BabyLM data. The pre-training goal is the regular next-token prediction goal of autoregressive LMs. We then post-train the models for three epochs with a higher learning rate (5e-4, to facilitate faster convergence) on the LMentry data. The complete LMentry datasets comprise 700k tokens (English) and 500k tokens (German), with task-specific counts (for 2.4k training examples) ranging from roughly 20k tokens for, e.g., the `first letter` tasks to 48k tokens for the `most associated word` task.

The post-training goal is also regular next-token prediction, but on the question-answer pairs found in LMentry, to induce this type of structure in our BabyLMs. For further details and to ensure full reproducibility, we also share our full pre- and post-training code on GitHub. While pre-training ran for 4–5h per model, the full post-training run took an additional 2.5h per model.

## B Data comparison

Table 3 presents the differences in dataset size between our models and the other "small" models evaluated in this study. With regard to pre-training tokens, the comparison models are trained on data that is many orders of magnitude larger than our BabyLM datasets. Furthermore, our datasets only comprise genuine language data, whereas the SmolLM2 and Llama-3.2 pre-training corpora also incorporate scientific papers, code or educational textbooks (cf. Allal et al., 2025 and Meta AI, 2024).

With regard to post-training data, information becomes more sparse. Whereas there is no comprehensive information of Meta's post-training recipe for the Llama-3.2 models, the SmolLM2 model family is post-trained on a vast array of datasets that optimize for chat completions, instruction-following, text rewriting, summarization, function calling, and also math data (Feng et al., 2024; Allal et al., 2025).

## C Tokenizer analysis

We use two standard measures, **fertility** and **PCW** (percentage of continued words), to evaluate the compression of our own tokenizers and of our comparison models' tokenizers (cf. Rust et al., 2021; Penedo et al., 2025). We calculate the measures as follows:

$$fertility = \frac{n_{subword\ tokens}}{n_{lexical\ words}}$$

$$PCW = \frac{n_{words\ split\ into\ more\ than\ one\ token}}{n_{lexical\ words}}$$

We measure fertility and PCW for two test datasets of 100,000 words per language, one sampled from our pre-training data (English/German BabyLM), and one sampled from our post-training data (English/German LMentry). Table 4 displays these results for our own tokenizers as well as the tokenizers from the SmolLM2 and Llama-3.2 model families (all different model sizes share the same tokenizer).

**Self-trained vs. off-the-shelf tokenizers** Generally, our self-trained BPE tokenizers feature a better compression than those from SmolLM2 and Llama-3.2. Despite our tokenizer being smaller (16k tokens vs. 50k/128k), words are generally split into fewer tokens and the percentage of continued words is also lower. While this is not surprising for the German data (where our tokenizer is the only monolingual one), the efficiency of our tokenizer for the English data is unexpected due to its smaller vocabulary size and therefore less words being stored holistically. One reasonable explanation could be a better fit to the data, as the subword vocabularies were induced over the pre-training data.

| | Data size (pre-training) | Data size (post-training) |
|---|---|---|
| Our BabyLMs | 10M words | 700k (Eng.), 500k (German), 2.4k examples |
| SmolLM2-135M | 600B tokens | 256k examples |
| SmolLM2-360M | 600B tokens | 256k examples |
| SmolLM2-1.7B | 1T tokens | 256k examples |
| Llama-3.2-1B | 9T tokens | Not available |
| Llama-3.2-3B | 9T tokens | Not available |

Table 3: Dataset sizes for our evaluated models.

| Tokenizer | Vocab. size | BabyLM (pre-train) Fertility | PCW | LMentry (Post-train) Fertility | PCW |
|---|---|---|---|---|---|
| English | | | | | |
| Our BPE | 16,000 | 1.45 | 0.32 | 1.66 | 0.39 |
| Our Char | 76 | 4.39 | 0.95 | 4.61 | 0.99 |
| SmolLM2 | 49,152 | 1.68 | 0.41 | 1.72 | 0.43 |
| Llama-3.2 | 128,256 | 1.58 | 0.39 | 1.63 | 0.41 |
| German | | | | | |
| Our BPE | 16,000 | 1.41 | 0.29 | 1.76 | 0.43 |
| Our Char | 76 | 5.03 | 0.98 | 5.37 | 0.99 |
| SmolLM2 | 49,152 | 2.31 | 0.68 | 2.57 | 0.74 |
| Llama-3.2 | 128,256 | 2.07 | 0.63 | 2.26 | 0.72 |

Table 4: Tokenizer quality measures for our evaluated models.

**Pre-training vs. post-training data** For all tokenizers, fertility and PCW are lower for the pre-training data and higher for the post-training data. Again, this is not overly surprising, as tokenizer vocabularies are induced over large text corpora and not the task data that LMentry contains. Interestingly, the division between our self-trained BPE tokenizer and the SmolLM2/Llama-3.2 tokenizers even hold true for the post-training data. Despite not being trained on this kind of data, our 16k subword vocabulary induced over the BabyLM data leads to better compression of LMentry data.

**Subword vs. character level** While subword vocabularies are induced over vast corpora with the BPE algorithm (Gage, 1994; Sennrich et al., 2016), the vocabulary of our character tokenizer is limited to 76 unique ASCII characters. As such, fertility and PCW are comparatively high: English words are split into 4–5 tokens, German words into more than 5 tokens on average. Almost all words are continued words (only one-character words like *I* or *a* are not split further, as they already consist of only one singular character). These results emphasize the trade-off between text compression and model capabilities. Although our character models compress text way worse than BPE models (and are therefore less suited to efficient autoregressive language modeling), this dip in compression power is needed for LMentry performance to improve.

## D Free-form generation

We also attempted free-form generation of answers, without constraining outputs to the task-specific possible answers. However, this yielded no usable results, as the resulting outputs turned out to be too messy to be easily evaluated. As manual evaluation of all answers is out of scope in light of the sheer number of task-specific questions, we attempted to automatically check the generated outputs by testing if the correct answer occurs somewhere in the output. Although this yielded impressive accuracies (e.g. boosting the BPE models to consistent 90% across the majority of tasks), a closer manual inspection of the responses showed that most free-form generations simply repeat the patterns/sentences of the input, and therefore contain all possible answers anyway. As of now, our models are not capable of useful free-form generation and need to be constrained with `outlines`.

We also share the full free-form outputs for our models here to allow further inspection and experimentation.

## E Example questions and number of answers for LMentry

In this section we briefly list all tasks included in LMentry, exemplary questions (in English) and the number of possible answers for each individual question type (which work the same for the German and English data).

- `smaller number` → 2 options (number-1 or number-2) | positive integers between 0–999
  Example: *Which number is smaller, 101 or 723?*
- `bigger number` → 2 options (number-1 or number-2) | positive integers between 0–999
  Example: *Which number is bigger, 487 or 110?*

- `last letter` → n options where n is the number of unique letters in the word (letter-1 or letter-2 or . . . or letter-n)
  Example: *What is the last letter of the word "beach"?*
- `first letter` → n options where n is the number of unique letters in the word (letter-1 or letter-2 or . . . or letter-n)
  Example: *What is the first letter of the word "decide"?*
- `more letters` → 2 options (word-1 or word-2)
  Example: *Which word has more letters, "fit" or "rice"?*
- `more letters length diff 1` → 2 options (word-1 or word-2)
  Example: *Which word has more letters, "honey" or "sock"?*
- `more letters length diff 3plus` → 2 options (word-1 or word-2)
  Example: *Which word has more letters, "honey" or "to"?*
- `less letters` → 2 options (word-1 or word-2)
  Example: *Which word has fewer letters, "no" or "chat"?*
- `less letters length diff 1` → 2 options (word-1 or word-2)
  Example: *Which word has fewer letters, "family" or "white"?*
- `less letters length diff 3plus` → 2 options (word-1 or word-2)
  Example: *Which word has fewer letters, "uniform" or "want"?*
- `word before` → n options where n is the number of words in the sentence (word-1 or word-2 or . . . or word-n)
  Example: *In the sentence "John was seeing his children", which word comes right before the word "was"?*
- `word after` → n options where n is the number of words in the sentence (word-1 or word-2 or . . . or word-n)
  Example: *In the sentence "We were glad it was over", which word comes right after the word "were"?*
- `first word` → n options where n is the number of words in the sentence (word-1 or word-2 or . . . or word-n)
  Example: *What is the first word of the sentence "He has left"?*
- `last word` → n options where n is the number of words in the sentence (word-1 or word-2 or . . . or word-n)
  Example: *What is the last word of the sentence "Donna fixed a sandwich for me"?*
- `first alphabetically` → 2 options (word-1 or word-2)
  Example: *In an alphabetical order, which of the words "at" and "policewoman" comes first?*
- `first alphabetically same first letter` → 2 options (word-1 or word-2)
  Example: *In an alphabetical order, which of the words "theirs" and "they" comes first?*
- `first alphabetically consecutive first letter` → 2 options (word-1 or word-2)
  Example: *In an alphabetical order, which of the words "case" and "duck" comes first?*
- `first alphabetically far first letter` → 2 options (word-1 or word-2)
  Example: *In an alphabetical order, which of the words "bake" and "zoo" comes first?*
- `first alphabetically different first letter` → 2 options (word-1 or word-2)
  Example: *In an alphabetical order, which of the words "your" and "centimeter" comes first?*
- `rhyming word` → 2 options (word-1 or word-2)
  Example: *Which word rhymes with the word "declare", "beer" or "wear"?*
- `rhyming word orthographically different` → 2 options (word-1 or word-2)
  Example: *Which word rhymes with the word "peak", "technique" or "steak"?*
- `rhyming word orthographically similar` → 2 options (word-1 or word-2)
  Example: *Which word rhymes with the word "child", "green" or "mild"?*
- `least associated word` → 5 options (word-1 or word-2 or word-3 or word-4 or word-5)
  Example: *Of the words "sweater", "shirt", "desk", "pants", and "socks", what is the word least associated with "clothes"?*
- `most associated word` → 5 options (word-1 or word-2 or word-3 or word-4 or word-5)
  Example: *Of the words "pear", "tram", "pineapple", "lemon", and "mango", what*

*is the word most commonly associated with "vehicles"?*

## F Additional models

As further ablations, we evaluate the non-LMentry-tuned models from the SmolLM2 suite (Allal et al., 2025), both in their base and instruction-tuned variants. Additionally, we also evaluate the base and instruction-tuned variant of the even larger Llama-3.2-3B model. Results are displayed in Tables 5 and 6.

**Comparison against our models** Neither for English nor for German LMentry do the off-the-shelf versions of small models outperform our character-based BabyLMs. Instead, their performance is rather comparable to our post-trained subword models. Only when directly post-trained on LMentry is this performance gap closed. This suggests that at BabyLM pre-training scale, adequate character tokenization can substitute for a lack of appropriate post-training. Although one could plausibly hypothesize that our subword-level BabyLMs are constrained in their LMentry performance by too little data or too few parameters, a dramatic increase in model size (to 3B parameters for Llama-3.2) and in pre-training data (cf. Appendix B) is still not sufficient to improve performance meaningfully; only direct post-training on LMentry can do so (as shown in Table 2).

**Base models vs. instruction-tuned models** Between base models and instruction-tuned models, again no clear differences emerge. Despite being post-trained on a wide variety of diverse tasks, e.g. text rewriting, summarization or function calling (Feng et al., 2024), the instruction-tuned models do not consistently outperform base models on LMentry. While the English results (Table 5) show a slight tendency towards better scores for instruction-tuned model variants, the German results (Table 6) are less straightforward. Only for the largest evaluated model, Llama-3.2-3B, clear advantages for the instruction-tuned models are visible.

**Scaling behavior** Finally, scaling behavior is not consistent between the two model families we analyzed. For the SmolLM models, the differences between the smallest model (135M, pre-trained on 600B tokens) and the largest model (1.7B, pre-trained on 1T tokens) are negligible. Across both English and German data, all models reach highest scores for some individual tasks. The picture is somewhat different for the Llama models, where the 3B variant (Tables 5 and 6) sometimes outperforms the 1B variant (Table 1) for the English data and frequently outperforms it for the German data (despite featuring the same pre-training data). In sum, the effect of a higher parameter count remains elusive. While it has no influence on the SmolLM2 models, it improves the capabilities of Llama-3.2. As the SmolLM2 family offers no larger model size than 1.7B, it could also be the case that the phase transition needed for such improvements occurs somewhere between SmolLM2's 1.7B parameters and Llama-3.2's 3B parameters.

| Task | English | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | SmolLM2-135M | | SmolLM2-360M | | SmolLM2-1.7B | | Llama-3.2-3B | |
| | Base | Instruct | Base | Instruct | Base | Instruct | Base | Instruct |
| smaller number | 0.66 | 0.60 | 0.67 | 0.67 | 0.65 | **0.68** | 0.55 | 0.61 |
| bigger number | 0.32 | 0.37 | 0.33 | 0.28 | 0.34 | 0.32 | 0.46 | **0.62** |
| last letter | 0.18 | 0.18 | 0.22 | 0.20 | 0.22 | 0.26 | 0.21 | **0.52** |
| first letter | 0.25 | 0.33 | 0.23 | **0.35** | 0.22 | 0.20 | 0.26 | 0.34 |
| more letters | 0.53 | 0.53 | 0.53 | 0.51 | 0.53 | 0.54 | 0.50 | **0.58** |
| more letters length diff 1 | 0.52 | **0.58** | 0.52 | 0.52 | 0.53 | 0.54 | 0.51 | 0.54 |
| more letters length diff 3plus | 0.56 | 0.58 | 0.52 | 0.56 | 0.52 | 0.55 | 0.58 | **0.69** |
| less letters | 0.45 | 0.47 | 0.48 | 0.48 | 0.46 | 0.48 | 0.45 | **0.49** |
| less letters length diff 1 | 0.49 | 0.46 | 0.47 | 0.44 | 0.48 | 0.47 | **0.53** | 0.42 |
| less letters length diff 3plus | 0.46 | 0.42 | 0.46 | **0.49** | 0.46 | 0.47 | 0.45 | 0.43 |
| word before | 0.17 | **0.23** | 0.15 | 0.18 | 0.10 | 0.18 | 0.16 | 0.18 |
| word after | 0.17 | 0.27 | 0.18 | 0.21 | 0.15 | 0.22 | 0.14 | **0.31** |
| first word | 0.23 | 0.12 | 0.20 | 0.20 | 0.17 | 0.10 | 0.24 | **0.35** |
| last word | 0.15 | 0.22 | 0.17 | 0.18 | 0.13 | 0.18 | 0.15 | **0.61** |
| first alphabetically | 0.58 | **0.65** | 0.60 | 0.53 | 0.55 | 0.61 | 0.42 | 0.51 |
| first alphabetically same first | 0.45 | 0.47 | 0.42 | 0.45 | 0.44 | 0.44 | **0.50** | 0.44 |
| first alphabetically cons. first | 0.53 | 0.50 | 0.48 | 0.44 | 0.53 | **0.54** | 0.52 | 0.50 |
| first alphabetically far first | **0.74** | 0.70 | 0.68 | 0.48 | 0.66 | 0.69 | 0.53 | 0.47 |
| first alphabetically diff. first | **0.65** | 0.63 | 0.62 | 0.50 | 0.63 | 0.62 | 0.48 | 0.50 |
| rhyming word | 0.51 | 0.46 | 0.49 | **0.53** | 0.48 | 0.48 | 0.50 | **0.53** |
| rhyming word ortho. different | 0.47 | 0.46 | 0.50 | 0.51 | 0.49 | 0.49 | 0.46 | **0.52** |
| rhyming word ortho. similar | 0.51 | 0.47 | **0.52** | 0.48 | **0.52** | 0.51 | 0.49 | **0.52** |
| least associated word | 0.22 | 0.16 | 0.19 | 0.20 | 0.22 | 0.23 | 0.22 | **0.52** |
| most associated word | 0.20 | 0.16 | 0.19 | 0.22 | 0.22 | 0.27 | 0.22 | **0.47** |
| MultiBLiMP | 0.97 | 0.95 | 0.97 | 0.96 | **0.99** | 0.98 | **0.99** | 0.97 |

Table 5: Full accuracies for additional small language models on English LMentry benchmark, best results marked in **boldface**.

| Task | German | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | SmolLM2-135M | | SmolLM2-360M | | SmolLM2-1.7B | | Llama-3.2-3B | |
| | Base | Instruct | Base | Instruct | Base | Instruct | Base | Instruct |
| smaller number | **0.68** | 0.56 | 0.66 | 0.66 | 0.66 | 0.67 | 0.64 | 0.50 |
| bigger number | 0.35 | 0.43 | 0.36 | 0.35 | 0.36 | 0.33 | 0.42 | **0.50** |
| last letter | 0.15 | 0.26 | 0.20 | 0.21 | 0.20 | 0.21 | 0.16 | **0.28** |
| first letter | **0.25** | 0.13 | 0.22 | 0.33 | 0.20 | 0.18 | 0.22 | 0.21 |
| more letters | 0.51 | 0.45 | **0.56** | 0.52 | 0.51 | 0.52 | 0.49 | 0.55 |
| more letters length diff 1 | 0.49 | **0.51** | 0.49 | 0.46 | 0.49 | **0.51** | 0.50 | 0.50 |
| more letters length diff 3plus | 0.50 | 0.49 | 0.52 | 0.55 | 0.48 | 0.54 | 0.55 | **0.64** |
| less letters | 0.49 | **0.53** | 0.49 | 0.45 | 0.49 | 0.49 | 0.48 | 0.47 |
| less letters length diff 1 | 0.50 | 0.48 | 0.48 | 0.49 | **0.54** | 0.46 | 0.49 | 0.52 |
| less letters length diff 3plus | 0.48 | 0.48 | 0.45 | 0.42 | **0.50** | 0.49 | 0.42 | 0.39 |
| word before | **0.12** | 0.10 | 0.08 | **0.12** | **0.12** | 0.11 | 0.11 | 0.11 |
| word after | 0.11 | 0.10 | 0.11 | 0.11 | 0.12 | 0.15 | 0.08 | **0.16** |
| first word | 0.11 | 0.05 | 0.09 | 0.07 | 0.12 | 0.11 | 0.08 | **0.13** |
| last word | 0.10 | **0.14** | 0.10 | **0.14** | 0.10 | 0.08 | 0.13 | 0.13 |
| first alphabetically | 0.62 | **0.64** | 0.59 | 0.55 | 0.57 | 0.57 | 0.46 | 0.54 |
| first alphabetically same first | **0.53** | 0.48 | **0.53** | 0.52 | 0.51 | 0.49 | 0.52 | 0.49 |
| first alphabetically cons. first | **0.56** | 0.50 | 0.52 | 0.48 | 0.50 | 0.45 | 0.45 | 0.48 |
| first alphabetically far first | 0.57 | 0.70 | 0.46 | 0.43 | 0.53 | **0.77** | 0.19 | 0.60 |
| first alphabetically diff. first | 0.60 | **0.73** | 0.46 | 0.48 | 0.63 | 0.72 | 0.24 | 0.54 |
| MultiBLiMP | 0.71 | 0.72 | 0.77 | 0.75 | 0.87 | 0.85 | **0.95** | 0.93 |

Table 6: Full accuracies for additional small language models on German LMentry benchmark, best results marked in **boldface**.