From Data to Knowledge: Evaluating How Efficiently Language Models Learn Facts

Daniel Christoph[†] **Max Ploner** [†‡] **Patrick Haller** [†] **Alan Akbik** [†‡]

[†] Humboldt-Universität zu Berlin
[‡] Science Of Intelligence
`<firstname>.<lastname>@hu-berlin.de`

Abstract

Sample efficiency is a crucial property of language models with practical implications for training efficiency. In real-world text, information follows a long-tailed distribution. Yet, we expect models to learn and recall frequent and infrequent facts. Sample-efficient models are better equipped to handle this challenge of learning and retaining rare information without requiring excessive exposure. This study analyzes multiple models of varying architectures and sizes, all trained on the same pre-training data. By annotating relational facts with their frequencies in the training corpus, we examine how model performance varies with fact frequency. Our findings show that most models perform similarly on high-frequency facts but differ notably on low-frequency facts. This analysis provides new insights into the relationship between model architecture, size, and factual learning efficiency.

1 Introduction

With the continued advancement of language models (LMs), comparing different architectures across various tasks and evaluating their performance using appropriate metrics becomes increasingly essential. These comparisons offer valuable insights into each architecture's general strengths and limitations. Sample efficiency is a key property of LMs, as sample-efficient models require less training and are thus more cost-effective (Micheli et al., 2023). As the LM processes large text corpora during pre-training, we are interested in assessing how efficiently each model learns specific relational facts comprising a subject, relation, and object.

A core question in this context is how different architectures handle the challenge of learning and retaining rare versus frequent facts. If two models are trained on the same dataset, their sample efficiency can be assessed by determining how often a fact must appear before each model successfully learns it (Botvinick et al., 2019; Liu et al., 2023). Models that rely predominantly on frequent facts while struggling with rarer ones—an issue caused by the long-tailed distribution of information in natural text (Zhang et al., 2024)—are considered sample-inefficient. Conversely, sample-efficient models should achieve higher accuracy on rare facts while maintaining strong performance on more common ones. To assess a model's factual knowledge, we use the BEAR probe (Wiland et al., 2024), which evaluates the model's ability to recall factual information across a wide range of subject-relation-object triples.

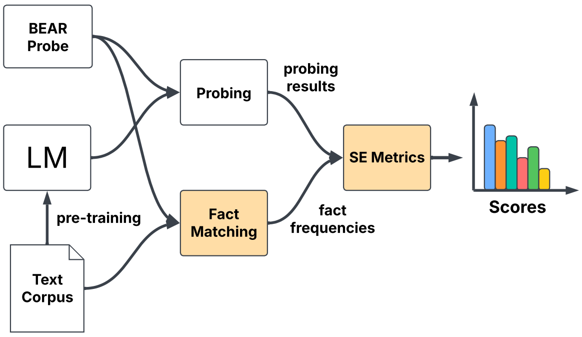


Figure 1: Sample efficiency evaluation of LMs.

An LM's factual knowledge can be probed by passing statements into the model (e.g., "*The capital of Germany is ...*") and evaluating its output to determine the represented knowledge of an LM (Roberts et al., 2020; Kalo and Fichtel, 2022; Kandpal et al., 2023). BEAR enables evaluation of both causal and masked LMs by constructing multiple answer choices, where each instance is transformed into a set of natural language statements: One for each answer option (e.g., *"Berlin"*, *"Paris"*, *"Buenos Aires"*, etc. for the relation HAS-CAPITAL and the subject *"Germany"*). The LM assigns log-likelihood scores to these statements, which are then ranked to determine the predicted answer.

Since BEAR contains no information about the pre-training data, it alone cannot be used to assess

Proceedings of the First Workshop on Large Language Model Memorization (L2M2), pages 29–46
August 1, 2025 ©2025 Association for Computational Linguistics

the sample efficiency.To address this, we need to not only determine whether the LM can correctly recall a given fact but also how many times it encountered it during pre-training (in the following, we call these “frequencies”). To create a correct sample efficiency evaluation procedure, we require an approach to estimate frequencies of facts from BEAR within a text corpus used for pre-training (see Figure 1). For this study, we employ a simple matching-based heuristic (see Section 3.1). Though unable to capture every occurrence of a fact, we assume it to be sufficiently accurate to predict the relative frequencies.

Given the information about how often an LM has encountered specific facts and whether it can recall them correctly, we must determine how to translate these fact-level data to a sample efficiency measure. Rather than estimating the point at which an LM transitions from not knowing to having learned the fact, we propose a more nuanced perspective: Measuring the incremental gain in factual knowledge as a function of the number of training samples. To operationalize this, we introduce two complementary metrics, which we use to quantify and compare the sample efficiency of different models over varying levels of fact exposure.

Contributions. Our contributions can be summarized as follows. We

1. Develop a framework to measure fact frequencies in text corpora efficiently and release counts for matched fact frequencies for a pre-training corpus,[1]

2. Propose a novel method for estimating sample efficiency using a model’s prediction on factual questions given the number of supporting frequencies in the pre-training corpus and

3. Compare models of three different architectures and varying sizes regarding their sample efficiency.

2 Related Work

Knowledge Probing. Petroni et al. (2019) introduced the influential *LAMA* probe, which evaluates language models by generating sentences that express factual relations, masking the object entity, and prompting the model to fill in the blank. This method, however, only supports single-subword token predictions and is not compatible with non-masked models like GPT. Variants adapted for causal (autoregressive) language models exist (Roberts et al., 2020; Kalo and Fichtel, 2022; Kandpal et al., 2023), but these cannot be used with masked LMs. To bridge this gap, BEAR (Wiland et al., 2024) reformulates relation instances into multiple-choice items, creating natural language statements for each candidate answer, and probing the model to assign log-likelihoods to each of the statements. By comparing the statements with the highest likelihood with the true answer enables evaluation across both model types (see Figure 2).

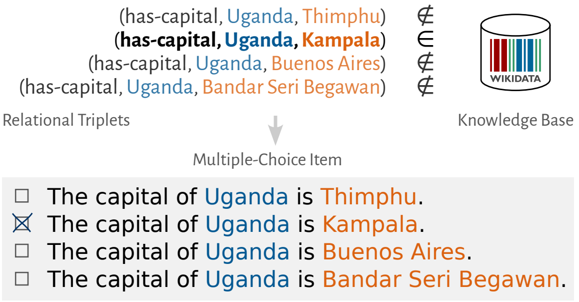

(has-capital, Uganda, Thimphu) ∉
(has-capital, Uganda, Kampala) ∈
(has-capital, Uganda, Buenos Aires) ∉
(has-capital, Uganda, Bandar Seri Begawan) ∉
WIKIDATA
Relational Triplets
Knowledge Base
Multiple-Choice Item
☐ The capital of Uganda is Thimphu.
☒ The capital of Uganda is Kampala.
☐ The capital of Uganda is Buenos Aires.
☐ The capital of Uganda is Bandar Seri Begawan.

Figure 2: In BEAR, one statement per answer option is passed to the LM (here using the template: “The capital of [X] is [Y].” and the subject “Uganda”). The assigned sentence-level likelihoods are then used to rank the answer options (figure from Ploner et al., 2024).

Sample Efficiency. In the current literature, sample efficiency can be defined as the property of a model to achieve similar performance to comparable models on tasks while requiring less training data or achieving better results while training on the same data (Liu et al., 2023; Lin et al., 2024). Reducing training time or data requirements is especially important when extensive data collection is expensive or impractical, which is especially challenging in domains with naturally low sample efficiency, potentially limiting real-world applicability (Yu, 2018; Feng et al., 2024).

Neural Scaling Laws. Kaplan et al. (2020) show that the test data’s loss value depends on the pre-training data scale. Given that the model is sufficiently large and enough compute is available, it follows a power-law relationship, i.e. in a log-log plot the function appears roughly as a linear line with negative slope and can hence be modeled by a function of the form $y = x^{-k}$.

[1] The repository containing the fact frequencies and code can be found here: github.com/Jabbawukis/sample-efficiency-evaluation.

Subsequent studies extend these findings to transfer learning (Hernandez et al., 2021), rigorously test this hypothesis, provide practical guidelines for optimal model-to-pre-training dataset size ratios (Hoffmann et al., 2022), and propose methods for computing scaling laws using intermediate checkpoints (Choshen et al., 2024). Finally, Godey et al. (2024) identify power-law relationships related to encoded geographic knowledge and Lu et al. (2024), the most relevant to our study, examines model size and training time in fact memorization.

To our knowledge, no prior work has examined the direct relationship between fact frequencies in the pre-training data and the model's ability to recall these facts.

3 Approach

To evaluate a model's sample efficiency, we employ a three-step approach. We build on BEAR and extend the probe by collecting fact frequencies (see Section 3.1) for a given pre-training corpus. We then train several LMs on this corpus (Section 3.2). This way, we can estimate how often a model has encountered a specific fact during its pre-training (and at which point). In Section 3.3, we introduce two novel sample efficiency metrics which produce aggregated scores based on the model's response to each sample and the sample's frequency.

3.1 Corpus Fact Frequency Statistics

To estimate how often a certain fact appears in the pre-training data, we look at single sentences and detect wether the fact is likely to be expressed within the sentence. For simplicity, we only check if two entities (belonging to a specific fact triple) occur within the same sentence from the corpus. If so, we assume the relational fact is represented within the sentence (Mintz et al., 2009).

For example, given the sentence "*The* Boeing 747 *is a long-range wide-body airliner designed and manufactured by* Boeing Commercial Airplanes *in the United States [...]*", the occurrence of both entities "*Boeing 747*" and "*Boeing Commercial Airplanes*" can be observed and the two entities are assumed to be linked by the MANUFACTURER relation. The entity "*Boeing Commercial Airplanes*" in this example may also be referred to as simply "*Boeing*" or "*Boeing commercial airplanes*". Hence, it is crucial to account for potential aliases of entities and to discard case sensitivity. Once two relation entities have been identified within a sentence, the sentence is counted as a fact occurrence (see Figure 3).

We use rule-based lemmatization (for English language) and sentence-splitting (*Sentencizer*) functionality provided by the spaCy Python library (Honnibal and Montani, 2017). Lemmatization greatly improves the matching with the entity aliases. The approach is implemented in the `FactMatcherSimple` class in the repostory linked in the contributions.

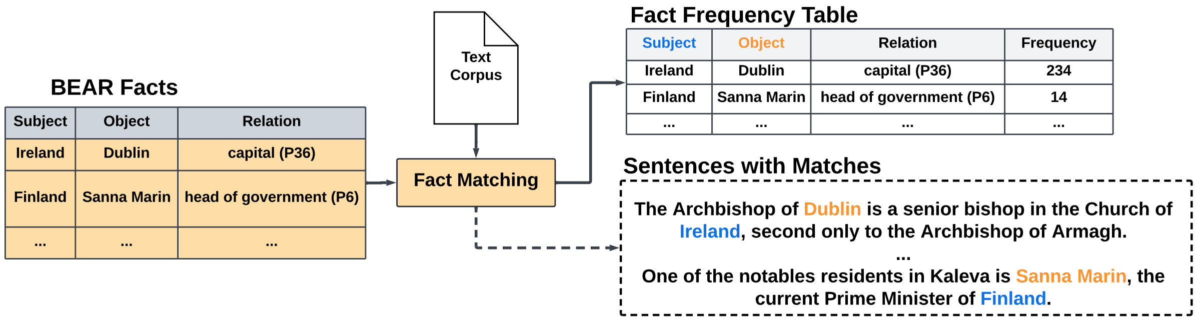
Selecting an appropriate corpus is crucial for generating useful fact-frequency statistics, as the chosen corpus must contain sufficient facts shared with the BEAR probe. If the text corpus lacks key information, entities from the BEAR probe may not appear with adequate frequency. To address this challenge, datasets derived from English Wikipedia articles, such as the Wikipedia dump language modeling dataset, can be utilized (Wikimedia Foundation, 2023). We applied this heuristic to the said corpus, and for better visualization, we placed each fact into a bucket relating to the overall frequency. The result is depicted in Figure 4.

3.2 Pre-Training the LMs

We pre-train several language model (LM) architectures and sizes, targeting comparable language modeling quality (see Section 4) on approximately five billion tokens of Wikipedia text (`20231101.en`; Wikimedia Foundation, 2023). For each model architecture, we train a small and a medium-sized model. To enable fine-grained fact tracking and to closely monitor each model's ability to recall facts over time, intermediate model checkpoints are saved and evaluated throughout training, allowing us to capture the learning dynamics in detail (see Section 4.2.1).

3.3 Evaluating the Sample Efficiency

To measure sample efficiency, a common approach is to track the number of encounters a model has with a specific fact during training and continuously probe the model to record when it has answered the question relating to the fact correctly (Liu et al., 2023; Lin et al., 2024). However, since facts are usually not learned in isolation, e.g., facts not associated with a specific question may still contain enough information to enable the model to acquire the knowledge required to answer the question correctly or make educated guesses, this approach may not suffice. Additionally, the model may provide



BEAR Facts

Subject	Object	Relation
Ireland	Dublin	capital (P36)
Finland	Sanna Marin	head of government (P6)
...	...	...

Fact Frequency Table

Subject	Object	Relation	Frequency
Ireland	Dublin	capital (P36)	234
Finland	Sanna Marin	head of government (P6)	14
...	...	...	...

Sentences with Matches

The Archbishop of Dublin is a senior bishop in the Church of Ireland, second only to the Archbishop of Armagh.

...

One of the notables residents in Kaleva is Sanna Marin, the current Prime Minister of Finland.

Figure 3: Example fact frequency table constructed from a text corpus. A fact is counted if the subject and the object occur within a sentence, even if the sentence does not explicitly express the relation.

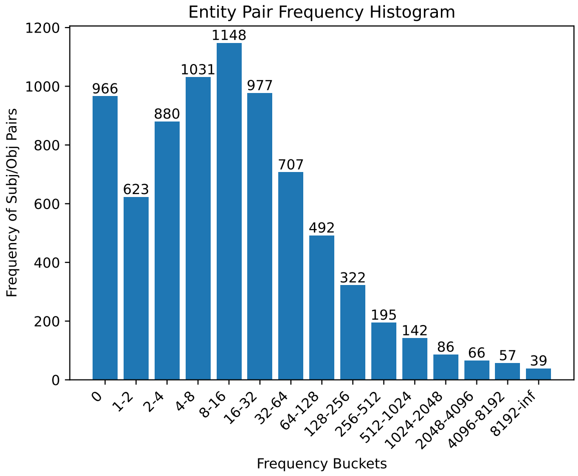


Figure 4: Number of matches for BEAR facts in the English Wikipedia dump (`20231101.en`; Wikimedia Foundation, 2023).

the correct answer at a specific moment in training but may later give the incorrect answer after it has processed more data, leading to a different outcome. There may not be a clear definition of learning a fact in a binary sense, as required.

To address these issues, we generalize this notion of sample efficiency: Instead of determining the critical point of knowledge acquisition, we conceptualize sample efficiency as the performance of correctly recalling facts as a function over the number of times the model has encountered this fact in the pre-training.

3.3.1 Weighted Accuracy Score on Frequency Buckets

A straightforward way is to measure the accuracy achieved on the facts of each frequency bucket (as illustrated in Figure 4). This provides a good initial impression of an LM's performance on rare and frequent facts. However, the array of scores makes it difficult to compare multiple models or track an LM's sample efficiency throughout the training. Hence, we propose an additional metric to condense these results to a single score, substantially simplifying the comparison. A computationally simple approach takes a weighted average over the buckets, weighting buckets with lower frequencies higher to focus on rarer facts. We propose the following weighting function based on the bucket i's lower bound l_i:

$$w_i = \begin{cases} \exp(-\lambda l_i), & \text{if } l_i \geq 1. \\ 0, & \text{otherwise.} \end{cases}$$

where λ is set to 0.05. The weight decreases with higher l_i, yielding $w_i \in [0, 1)$, resulting in a declining impact of the high-frequency facts on the overall weighted accuracy (see Appendix Figure 9a). The weighted accuracy is then calculated (with the accuracy score acc_i on bucket i) as:

$$\frac{1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} w_i \cdot \text{acc}_i$$

If the fact has a particular frequency of x, we assign the fact to the bucket with a lower bound of l_i and an upper bound of u_i iff. $x \in [l_i, u_i)$.

3.3.2 Modeling the Probability of an LM to Answer Correctly

A second approach is to apply a probabilistic interpretation and to treat sample efficiency as a key property of the function mapping the number of fact frequencies to the probability of the model recalling the fact accurately. Within this framework, the threshold of the step function would represent the conventional notion of sample efficiency: The exact number of frequencies needed to give the correct answer consistently.

The step function may be ill-suited to model the actual probability of the model giving the correct answer. Instead, we propose to use a continuous function, where a higher slope of the function indicates a higher likelihood of the model learning a function and, thus, a higher sample efficiency. This approach eliminates the need to identify when a model has learned a specific fact by generalizing the evaluation to groups of facts rather than individual instances, potentially allowing for a more robust assessment of sample efficiency across varying levels of exposure in the training data.

We statistically model the probability of an LM correctly answering a question, given the number of frequencies of the related fact in the training data using a power scaling function (see also the segment on neural scaling laws in Section 2; Kaplan et al., 2020):

$$F(x) = 1 - \left(L_0 + \frac{x_0}{(1+x)^{\alpha_m}} \right)$$

Here, x is the frequency of a fact, and L_0, x_0, and α are found by statistical fitting. While L_0 and x_0 are dataset dependent, there is one α_m per LM.

α_m controls the slope of the probability function: Higher values increase the probability per additional occurrence, indicating higher sample efficiency.

L_0 can be interpreted as the constant rate of error that is unavoidable, given the possibility that the BEAR probe contains errors (zero would indicate that the potential errors in the probe's question catalog do not influence the function).

x_0 is at least influenced by the fact-matching algorithm described in Section 3.1. Underestimating fact frequencies could result in a lower estimated x_0 value. Values lower than one indicate the LM's initial probability of correctly answering a fact can be ≥ 0, and values close to zero suggest an unexpectedly high probability, even though the fact frequency is zero. Such a value might be produced due to the simplicity of the fact-matching heuristic or the learning of facts through other facts that hold helpful information for the fact in question or, in other words, educated guesses.

Representing LM m's prediction on fact i as $T_{m,i}$ (one if the model answered correctly, zero otherwise) yields a likelihood $p_{m,i}$ that the model makes the given prediction (given the modeled probability):

$$p_{m,i} = T_{m,i} F(x_i) + (1 - T_{i,m})\,(1 - F(x_i))$$

The overall probability of the predictions occurring as they have given the parameters L_0, x_0, and α_m is then given by:

$$P\left(L_0, x_0, \boldsymbol{\alpha}\right) = \prod_m \prod_i p_{m,i}$$

We maximize the joint probability (by minimizing the negative log-likelihood) over all BEAR probe facts and models. This yields the maximum likelihood estimate for our dataset-specific parameters L_0, x_0, and model-specific α_m. LMs with a higher α_m value can be considered more sample efficient as they exhibit a higher increase in the probability of answering a factual item per observed sample.

4 Empirical Evaluation

Leveraging the proposed approach allows us to address the following questions: (1) which model architecture demonstrates higher levels of sample efficiency, (2) and how well a model recalls facts throughout the training.

LM Architecture Selection. Newer RNN-based architectures indicate advantages over transformer-based architectures in data-scarce scenarios and thus may indicate a higher sample efficiency (Haller et al., 2024). As the model architectures evaluated in this work consist of transformer-based GPT2 (Radford et al., 2019) and LLaMA (Touvron et al., 2023), RNN-based xLSTM (Beck et al., 2024) and state-space-based MAMBA2 (Dao and Gu, 2024), the selected model architectures are well-suited for this study and may contribute to a deeper understanding of sample efficiency, particularly in the context of RNNs versus transformers, as well as broader trends across different architectural paradigms.

We train two groups of models. A small group with sizes around 200 million parameters, and a medium-sized group with around 400 million parameters. Due to limited resources, we are restricted to a limited set of training runs and LM sizes.

For model pre-training of the different model architectures, we use the models and trainer implemented in the Hugging Face *transformers* library (Wolf et al., 2020).

4.1 Sample Efficiency of Different LM Architectures

Our first experiment compares the LMs' sample efficiency. Specifically, we evaluate the model's

	Model	#params	ACC	WASB	α_m
SMALL	GPT2	209M	28.0%	21.8%	0.084
	LLAMA	208M	**31.0%**	**24.1%**	**0.103**
	xLSTM	247M	28.1%	21.7%	0.086
	MAMBA2	172M	28.6%	22.9%	0.087
MEDIUM	GPT2	355M	30.4%	24.0%	0.098
	LLAMA	360M	**34.4%**	**27.9%**	**0.120**
	xLSTM	406M	30.7%	24.2%	0.100
	MAMBA2	432M	32.1%	26.2%	0.106

Table 1: Resulting measures for LM's after pre-training on the complete corpus.

	Model	#params	< 1024	≥ 1024
SMALL	GPT2	209M	26.2%	83.4%
	LLAMA	208M	**29.1%**	**88.7%**
	xLSTM	247M	26.4%	79.4%
	MAMBA2	172M	26.8%	82.2%
MEDIUM	GPT2	355M	28.6%	**87.5%**
	LLAMA	360M	**32.7%**	85.4%
	xLSTM	406M	29.0%	82.2%
	MAMBA2	432M	30.5%	81.4%

Table 2: Accuracy on high and low frequency facts on BEAR.

accuracy scores on each frequency bucket, apply the proposed metrics, and calculate the overall accuracy on all BEAR questions for comparison.

4.1.1 Experimental Setup

Each model is trained on the same information-rich text corpus (Wikimedia Foundation, 2023) using the same vocabulary (GPT2 tokenizer) and training parameters to ensure maximum comparability (see Appendix Table 3). Each pre-training run took two to three days and was done on a single NVIDIA A100 (80GB) GPU. The models were evaluated using the proposed sample efficiency metrics (see Section 3.3). Additionally, each model was evaluated using several tasks from the language model evaluation harness (Gao et al., 2024), including *winogrande*, *wsc273*, *lambada_standard* and *pile_10k* to test the model's general language modeling capabilities (see Appendix Table 7).

4.1.2 Results

Table 1 reports the overall accuracy on all questions (ACC), the weighted accuracy score on the frequency buckets (WASB, see Section 3.3.1), and the optimized α_m values (see Section 3.3.2) for the LMs in consideration (final state). The L_0 and x_0 values are optimized to 0.00 and 0.88, respectively. This indicates a base probability of a question being answered correctly by the model greater than zero and the general correctness of the BEAR probe question catalog.[2] Going forward, we propose using the values we determined since x_0 and L_0 are dataset characteristics and not model-dependent (though future refinements using a larger set of models are possible).

These results highlight two key observations. First, sample efficiency improves with increasing model size. Second, both LLAMA models consistently outperform other architectures with similar parameters.

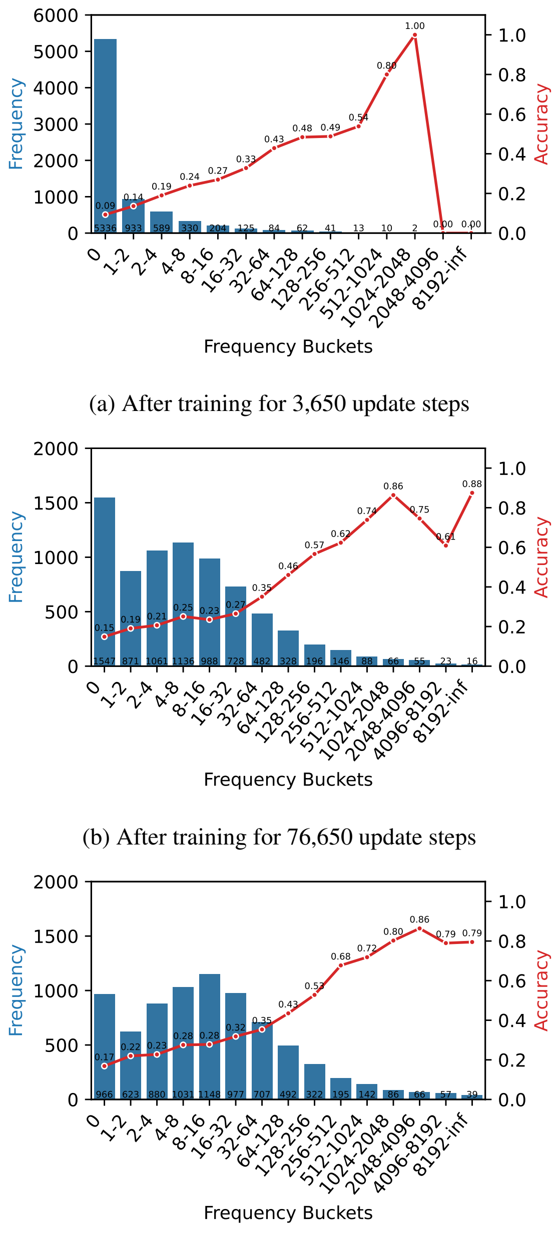
Accuracies on Frequency Buckets. Figure 16 in the appendix reports the model's accuracies on each frequency bucket. As Section 3.3.1 mentions, these scores provide an initial impression of the model's overall sample efficiency. Larger models achieve a higher accuracy score on the low to mid-frequency buckets (≤ 128). This finding indicates that larger LMs may learn less frequent facts better.

Accuracies on High Occurring Facts. To verify this hypothesis, we split the facts into *high-frequency* ($x \geq 1000$) and *low-frequency* ($x < 1000$) facts and measure the accuracy on each of the splits. Looking at these accuracies (in Table 2), we again observe an explicit ordering of the model performances in correlation with their size (as observed in Table 1) for low-frequency facts. However, the performance on high-frequency facts does not follow this trend.

Accuracies on high-occurring facts show less deviation between the models, as some small models achieve accuracy scores comparable to the medium models (e.g., small GPT2 and medium MAMBA2). These findings show that larger LMs may not memorize high-frequency, possibly redundant facts significantly better than smaller models, in line with observations made by Lu et al. (2024).

The results indicate that eliminating high-frequency facts or adjusting their influence on the final accuracy score to mitigate their impact may be necessary to measure sample efficiency effectively. This, however, may heavily depend on the dataset used for pre-training and may not always be required. In some cases, the accuracy alone may suffice to distinguish sample-efficient from sample-inefficient models (also see Figure 12 and 15 in the appendix).

[2] For BEAR-big, the resulting values for L_0 and x_0 are 0.0 and 0.92, respectively. The respective table (5) can be found in Appendix B.



(a) After training for 3,650 update steps

(b) After training for 76,650 update steps

(c) After training for 153,300 update steps

Figure 5: Accuracy on frequency buckets during training of Mamba2 with 432 million parameters. The top, middle, and bottom graphs depict the model's accuracy at the training's beginning, middle, and end.

4.2 Learning Dynamics

To investigate how the models acquire knowledge throughout the training, we probe the model periodically throughout the training. This also enables us to check if the proposed metrics are predictive of the final results: When do the bucket accuracies stabilize, and can we predict the final accuracy by extrapolating from a given checkpoint (knowing how often the facts will be seen in the data yet to be used during training)?

4.2.1 Experimental Setup

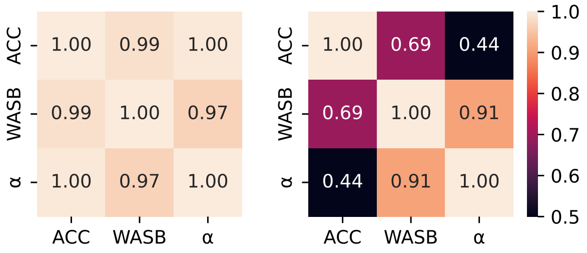
The dataset is shuffled to account for a possible unbalanced distribution of data point sizes and was divided into 42 slices with 3650 steps per slice, with a train batch size of 32, gradient accumulation set to 8, and 934,840 rows per slice after tokenization on average ($934,840 \approx 8 \times 32 \times 3650$). Each slice is then processed using the fact-matching heuristic. We calculate the average[3] number of training steps performed for each slice and save the model's state after a slice has been processed. Each state is then individually probed and evaluated based on the number of facts with specific frequencies the model has seen up until then. Probing each checkpoint for a single training run (i.e., 42 different model states) using BEAR-big (which includes BEAR as a subset) took approximately one day (single NVIDIA A100 (80GB) GPU). To substantially cut down the probing time, we recommend probing only using BEAR (without BEAR-big) and fewer checkpoints in practical settings.

4.2.2 Results

During training, we observe a gradual convergence toward specific accuracy scores for the lower frequency buckets relatively early, with increasingly smaller changes in the later stages of training. This indicates that a model's ability to learn a fact improves with the general learning of the meaning of language but remains relatively stagnant concerning frequency. This behavior is depicted in Figure 5 (accuracy scores on frequency buckets during training of MAMBA2 with 432 million parameters and probed with BEAR).

Looking at the weighted accuracy scores (see Section 3.3.1) and α-values (see Section 3.3.2) of the LMs over each slice, we observe a similar trend, with each model reaching a specific score early in training, with relatively minimal changes in the later stages of training (see Appendix Figure 10 and 11). However, the degree of increase in the scores during training seems to depend on the model's overall capability to learn facts, as models with a higher final α-value and weighted accuracy score show steeper increases, only reaching a stagnation point later in training.

[3] Using the mean instead of the slice-dependent number is not entirely accurate. However, since the variation between the slices (regarding training steps) is minimal, this simplification should not change the results.



(a) Final scores (b) Over all slices

Figure 6: Correlation matrix of the final scores and over all slices.

4.2.3 Correlation Between The Metrics

The proposed metrics indicate a clear trend: Larger models tend to outperform smaller models and are thus more sample-efficient, with exceptions observed in the LLAMA models, where the smaller model demonstrates competitive or superior performance compared to larger RNN-based models. This highlights the role of architectural efficiency beyond just scale. Additionally, the progression of the scores of each state of the models follows a similar trajectory in both proposed metrics, with minor variations in magnitude and fluctuations at specific points (see Appendix Figures 10 and 11). This similarity suggests that both metrics are valid model performance indicators and can be used interchangeably or individually to assess sample efficiency. This results in a high correlation[4] between the proposed metrics across slices, while the correlation with the general accuracy is lower in comparison (see Figure 6b). On the other hand, we observe strong positive correlations for each metric for the final state (see Figure 6a), as each metric sorts the model's final measurement similarly (larger models outperform smaller ones).

4.3 Metric Robustness

To further investigate the metrics' robustness to changes in the testing dataset's composition, we create two splits with 1000 facts from BEAR, each with a different frequency profile. Using these two splits, we aim to determine the impact of the different frequency profiles on the final metric.

Ideally, any testing dataset (no matter the makeup) could be used to estimate a model's sample efficiency based on the response patterns and information about the fact frequencies. We hypothesize that the fact frequencies highly impact the raw accuracy over the facts. In contrast, the weighted accuracy (WASB) and the modeling-based sample efficiency metric α might be less influenced by the sampling of the splits.

It should be noted that this assumes that the samples across the datasets are (on average) equally hard: The probability of the model to correctly predict the fact *only* depends on the pre-training data and the model's sample efficiency (and not other difficulty factors).

4.3.1 Experimental Setup

For the *low-frequency* split, we sample 80% of the facts from facts with less than eight occurrences and the other 20% from facts with eight occurrences or more. We do the opposite in the *high-frequency* split (i.e., 80% from facts with eight occurrences or more). The threshold must be set sufficiently to guarantee a strong bias within the split towards facts with a desired frequency. Otherwise, the split would be too close to the original data set. This can be achieved by calculating the median bucket lower bound for the fact counts, functioning as said threshold. We evaluate the final checkpoints of each model on these two new datasets and compute the different metrics.

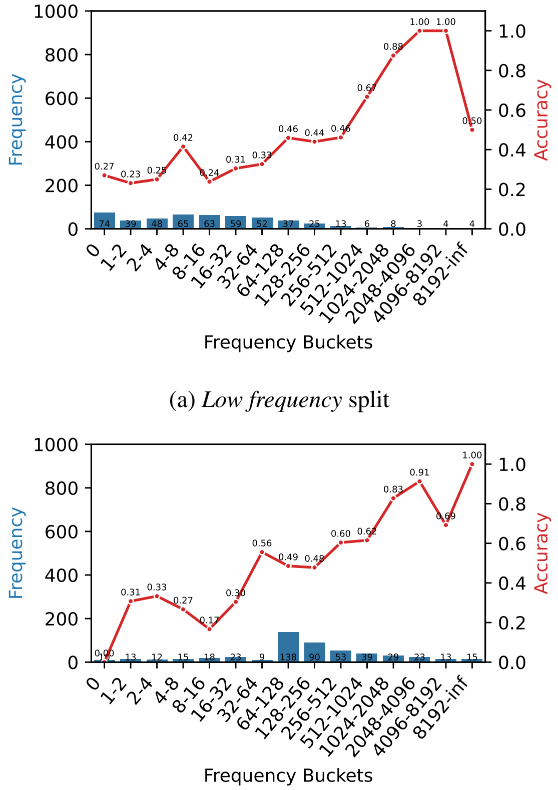
4.3.2 Results

The results are depicted in Figure 17 in the appendix. The exemplary resulting frequency histogram and the accuracy for each bucket for MAMBA2 are shown in Figure 7.

Accuracy. The variation in the general accuracy among the models in the frequency splits is substantial. Compared to the scores on the complete dataset, the accuracy is lower if primarily low-frequency samples are selected and considerably higher in the high-frequency split (see Appendix Figure 17).

Weighted Accuracy (WASB). For the weighted accuracy measure on the frequency buckets for each model, the variation between the low and high frequency splits remains lower than the general accuracy. However, the weighted accuracy approach is limited by the need to adjust the buckets' resolution as more facts produce more robust results. Further investigation is needed to determine if there are robust ways to set the boundaries of the buckets based on the fact frequencies and the weights

[4] Correlations were computed between metric scores across models at final training (Fig. 6a; raw scores in Table 1) using vectors $v_M \in \mathbb{R}^{m\times 1}$. Correlations across all 42 training slices (Fig. 6b) use flattened vectors $v_M \in \mathbb{R}^{m\times 42}$. Columns are sorted by correlation with overall accuracy.



(a) *Low frequency* split

(b) *High frequency* split

Figure 7: Accuracy on frequency buckets after training of MAMBA2 for the two splits.

of each bucket based on these boundaries. This may lead to more robust measures where every subsample of the dataset can be used to estimate the overall performance. Additionally, calculating the weighted accuracy using accuracies on frequency buckets may result in less reliable scores when the number of samples within a bucket is too low. To address this, incorporating a confidence coefficient can help adjust for the increased uncertainty associated with smaller sample sizes.

α-Sample Efficiency. The α-values exhibit the lowest variation between the low and high frequency splits (see Appendix Figure 17). Thus, this modeling-based metric provides the highest robustness against fact frequency changes, resulting in the most reliable measures.

5 Conclusion

We presented a sample efficiency evaluation framework that compares LMs' ability to learn facts given a text corpus and the BEAR probe. The framework consists of a fact-matching algorithm that extracts fact frequency statistics from a sizable data set and two sample efficiency metrics. We trained several state-of-the-art LMs in a controlled setting, ensuring the validity of the evaluation, and provided a detailed analysis of the different architecture results.

The performance on high-frequency facts indicates less divergence between models regarding size. In contrast, performance on low-frequency facts demonstrates the increased sample efficiency gained with model size. The proposed metrics are capable of identifying the superiority in sample-efficiency of the transformer-based LLAMA models, achieving the highest scores in all metrics, with the state-space-based MAMBA2 models closing behind.

The proposed metrics correlate strongly in respect to the final model stages as well as across the training. This indicates that a different property is measured than in raw accuracy. Additional experiments show, that the metrics are relatively robust to varying fact frequency distributions in pre-training data. We believe the plausibility of the design choices together with these findings make the metrics strong candidates for measuring sample efficiency.

Limitations

This work is limited to a simple fact-matching heuristic, as discussed in Section 3.1. This heuristic produces sufficiently accurate statistics and provides a high degree of flexibility; however, more advanced heuristics, e.g., adding natural language processing pipelines such as entity linking, could produce more accurate fact occurrence counts, as they potentially reduce the possible mismappings of entities due to likely ambiguity or relation misidentification. Furthermore, the proposed probability function lower bound depends on L_0, validated empirically in this work (see Section 4.1.2). However, this initial L_0 value can change depending on the correctness of the probe (or the training text corpus), as significant errors and noise can alter the outcome of the measurements. Thus, further research could be conducted on the robustness of the metric in those scenarios. Finally, this work is limited to evaluating models of small to medium size. Whether the observed trend of increasing sample efficiency with model size holds for larger models exceeding one billion parameters remains open.

Acknowledgments

We thank all reviewers for their valuable comments. Alan Akbik and Max Ploner are supported under Germany's Excellence Strategy "Science of Intelligence" (EXC 2002/1, project number 390523135). Further, Alan Akbik and Patrick Haller are supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Emmy Noether grant "Eidetic Representations of Natural Language" (project number 448414230).

References

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2024. xlstm: Extended long short-term memory. *Preprint*, arXiv:2405.04517.

Matthew Botvinick, Sam Ritter, Jane X. Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. 2019. Reinforcement Learning, Fast and Slow. *Trends in Cognitive Sciences*, 23(5):408–422.

Leshem Choshen, Yang Zhang, and Jacob Andreas. 2024. A Hitchhiker's Guide to Scaling Law Estimation. *Preprint*, arXiv:2410.11840.

Tri Dao and Albert Gu. 2024. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning (ICML)*.

Kehua Feng, Keyan Ding, Kede Ma, Zhihua Wang, Qiang Zhang, and Huajun Chen. 2024. Sample-efficient human evaluation of large language models via maximum discrepancy competition. *Preprint*, arXiv:2404.08008.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. A framework for few-shot language model evaluation.

Nathan Godey, Éric de la Clergerie, and Benoît Sagot. 2024. On the Scaling Laws of Geographical Representation in Language Models. *Preprint*, arXiv:2402.19406.

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.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. 2021. Scaling Laws for Transfer. *Preprint*, arXiv:2102.01293.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, and 3 others. 2022. Training Compute-Optimal Large Language Models. *Preprint*, arXiv:2203.15556.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Jan-Christoph Kalo and Leandra Fichtel. 2022. KAMEL : Knowledge Analysis with Multitoken Entities in Language Models. In *Automated Knowledge Base Construction*.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large Language Models Struggle to Learn Long-Tail Knowledge. *Preprint*, arXiv:2211.08411.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. *Preprint*, arXiv:2001.08361.

Jianghao Lin, Xinyi Dai, Rong Shan, Bo Chen, Ruiming Tang, Yong Yu, and Weinan Zhang. 2024. Large language models make sample-efficient recommender systems. *Preprint*, arXiv:2406.02368.

Nelson F. Liu, Ananya Kumar, Percy Liang, and Robin Jia. 2023. Are sample-efficient nlp models more robust? *Preprint*, arXiv:2210.06456.

Xingyu Lu, Xiaonan Li, Qinyuan Cheng, Kai Ding, Xuanjing Huang, and Xipeng Qiu. 2024. Scaling Laws for Fact Memorization of Large Language Models. *Preprint*, arXiv:2406.15720.

Vincent Micheli, Eloi Alonso, and François Fleuret. 2023. Transformers are sample-efficient world models. *Preprint*, arXiv:2209.00588.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Max Ploner, Jacek Wiland, Sebastian Pohl, and Alan Akbik. 2024. LM-PUB-QUIZ: A Comprehensive Framework for Zero-Shot Evaluation of Relational Knowledge in Language Models. *Preprint*, arXiv:2408.15729.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How Much Knowledge Can You Pack Into the Parameters of a Language Model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Wikimedia Foundation. 2023. Dump of English Wikipedia of November 1st, 2023.

Jacek Wiland, Max Ploner, and Alan Akbik. 2024. BEAR: A unified framework for evaluating relational knowledge in causal and masked language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2393–2411, Mexico City, Mexico. Association for Computational Linguistics.

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, and 3 others. 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.

Yang Yu. 2018. Towards sample efficient reinforcement learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5739–5743. International Joint Conferences on Artificial Intelligence Organization.

Chongsheng Zhang, George Almpanidis, Gaojuan Fan, Binquan Deng, Yanbo Zhang, Ji Liu, Aouaidjia Kamel, Paolo Soda, and João Gama. 2024. A systematic review on long-tailed learning. *Preprint*, arXiv:2408.00483.

A Pre-Training & Model Configuration

Parameter	Value
per_device_train_batch_size	32
gradient_accumulation_steps	8
num_train_epochs	1
weight_decay	0.1
warmup_steps	1000
lr_scheduler_type	cosine
learning_rate	5e-4
fp16	True

Table 3: Training Hyperparameters.

		Small	Medium
GPT2	Parameters	209M	355M
	Hidden Size	768	1024
	Intermediate Size	3072	4096
	Hidden Layers	24	24
	Num Heads	16	16
xLSTM	Parameters	247M	406M
	Hidden Size	768	1024
	Intermediate Size	2048	2731
	Hidden Layers	24	24
	Num Heads	4	4
MAMBA2	Parameters	172M	432M
	Hidden Size	768	1024
	Intermediate Size	1536	2048
	Hidden Layers	24	48
	Num Heads	24	32
	State Size	32	32
LLAMA	Parameters	208M	360M
	Hidden Size	768	960
	Intermediate Size	1536	2560
	Hidden Layers	36	32
	Num Heads	9	15

Table 4: Model configurations used during training.

B Further Results

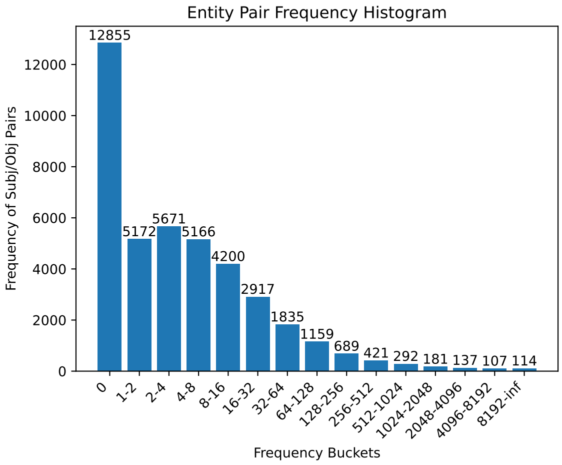


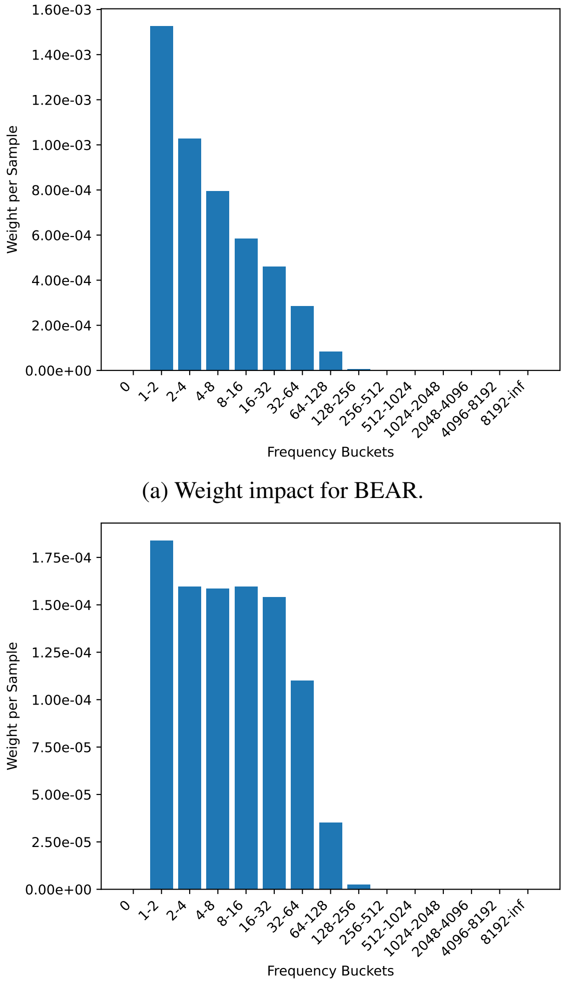
Figure 8: Number of matches for BEAR-big facts in the English Wikipedia dump (`20231101.en`; Wikimedia Foundation, 2023).

	Model	#params	ACC	WASB	α_m
SMALL	GPT2	209M	16.2%	16.0%	0.064
	LLAMA	208M	**18.2%**	**18.0%**	**0.079**
	xLSTM	247M	15.6%	15.6%	0.064
	MAMBA2	172M	16.1%	16.1%	0.064
MEDIUM	GPT2	355M	17.7%	17.5%	0.074
	LLAMA	360M	**20.1%**	**20.1%**	**0.091**
	xLSTM	406M	17.3%	17.0%	0.073
	MAMBA2	432M	18.5%	18.6%	0.080

Table 5: Results on BEAR-big.

	Model	#params	< 1024	≥ 1024
SMALL	GPT2	209M	15.3%	79.9%
	LLAMA	208M	**17.3%**	**83.8%**
	xLSTM	247M	14.8%	77.9%
	MAMBA2	172M	15.3%	77.3%
MEDIUM	GPT2	355M	16.8%	**82.1%**
	LLAMA	360M	**19.3%**	82.0%
	xLSTM	406M	16.4%	79.5%
	MAMBA2	432M	17.7%	79.4%

Table 6: Accuracy on high and low occurring facts on BEAR-big.



(a) Weight impact for BEAR.

(b) Weight impact for BEAR-big.

Figure 9: Impact of the frequency bucket weight per number of samples.

	Model	#params	winogrande	wsc273	lambada_standard acc	lambada_standard PPL	pile_10k PPL
SMALL	GPT2	209M	50.36% ± 1.4%	53.11% ± 3.03%	16.63% ± 0.52%	652.0058 ± 33.1575	14389.4299
	LLaMA	208M	50.59% ± 1.4%	55.68% ± 3.01%	15.58% ± 0.51%	694.1146 ± 34.3843	65059.5665
	xLSTM	247M	50.43% ± 1.4%	54.95% ± 3.02%	9.35% ± 0.41%	1536.1172 ± 74.8833	966.7574
	MAMBA2	172M	50.2% ± 1.4%	50.92% ± 3.03%	7.68% ± 0.37%	2183.7652 ± 109.3855	1295.2241
MEDIUM	GPT2	355M	51.62% ± 1.4%	54.58% ± 3.02%	16.44% ± 0.52%	592.8151 ± 29.6474	17984.4641
	LLaMA	360M	51.85% ± 1.4%	54.58% ± 3.02%	15.76% ± 0.51%	508.1769 ± 23.8731	216732.2782
	xLSTM	406M	51.46% ± 1.4%	50.55% ± 3.03%	11.97% ± 0.45%	739.1623 ± 34.8244	890.4901
	MAMBA2	432M	50.67% ± 1.4%	54.58% ± 3.02%	7.88% ± 0.38%	1594.1999 ± 77.5151	1116.7870

Table 7: LM Evaluation Harness Results.

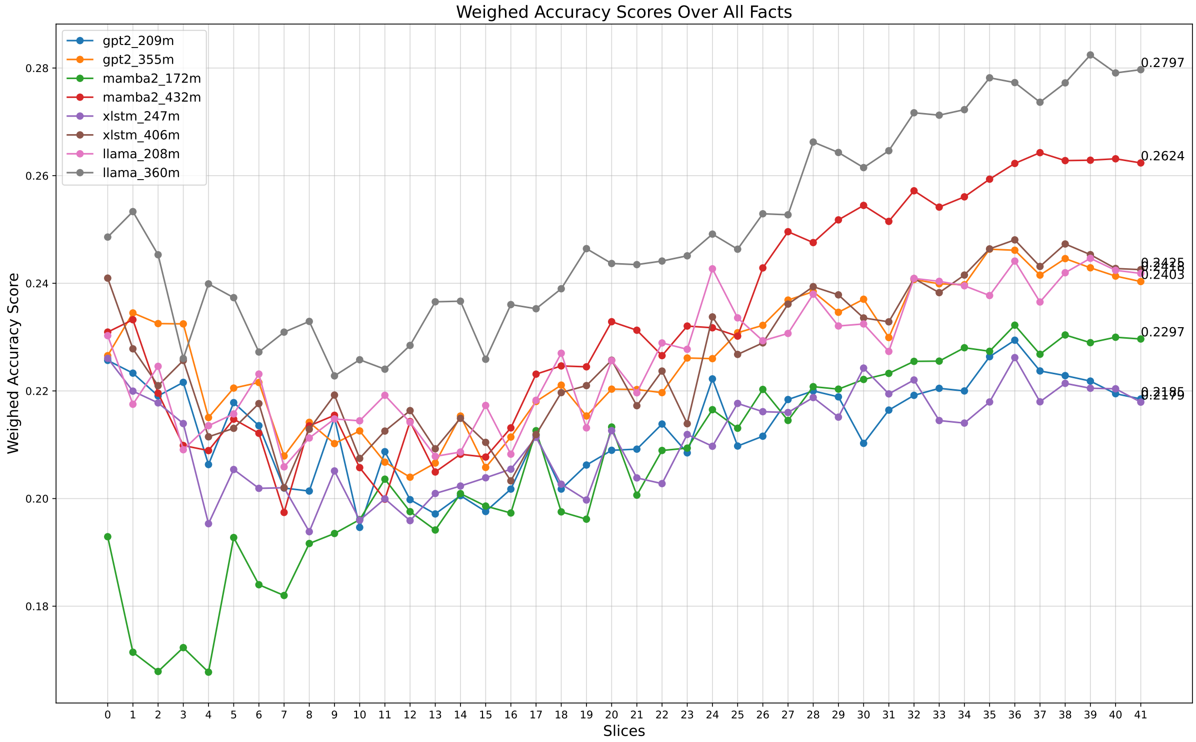


Figure 10: Development of the weighted accuracy (WASB) throughout the pre-training.

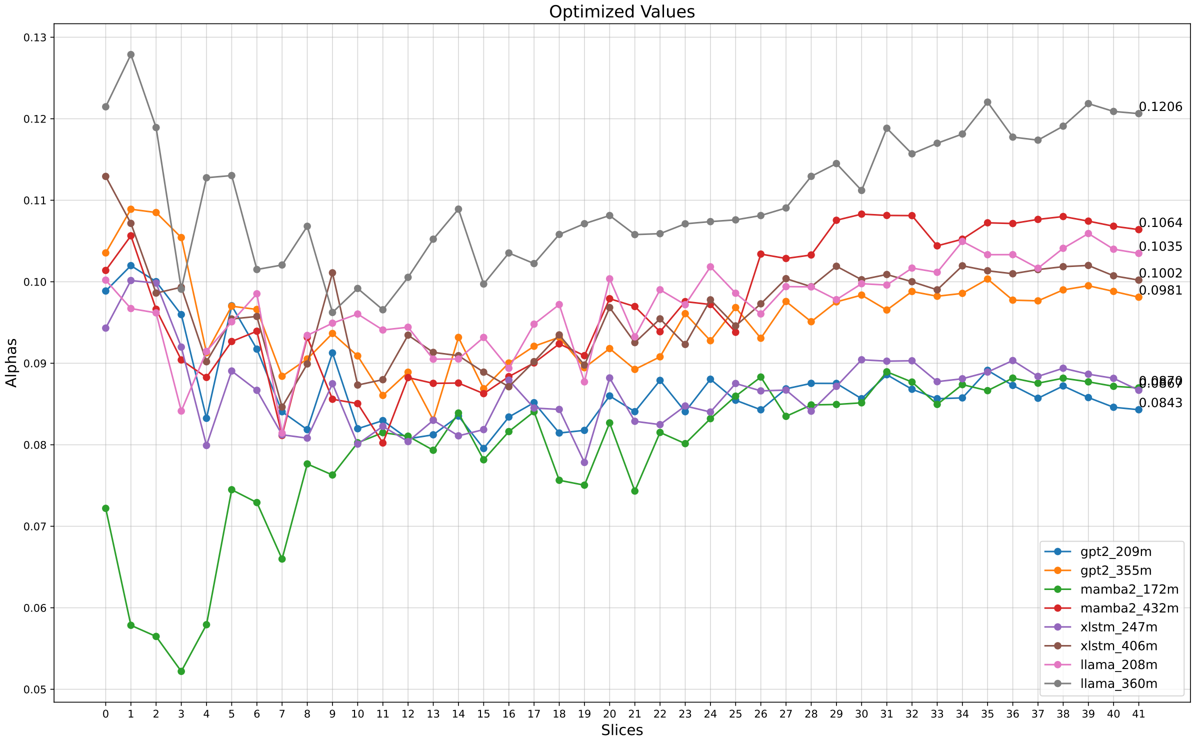


Figure 11: Development of α_m over the course of the pre-training.

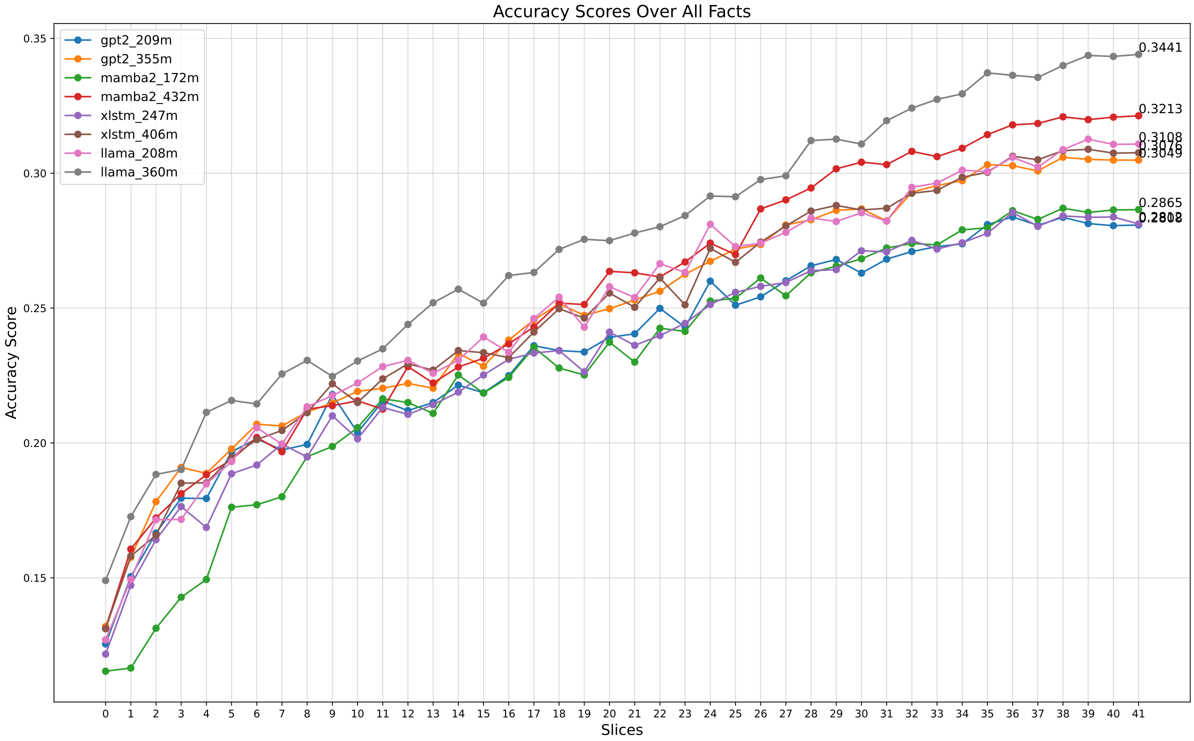


Figure 12: Development of the accuracy throughout the pre-training.

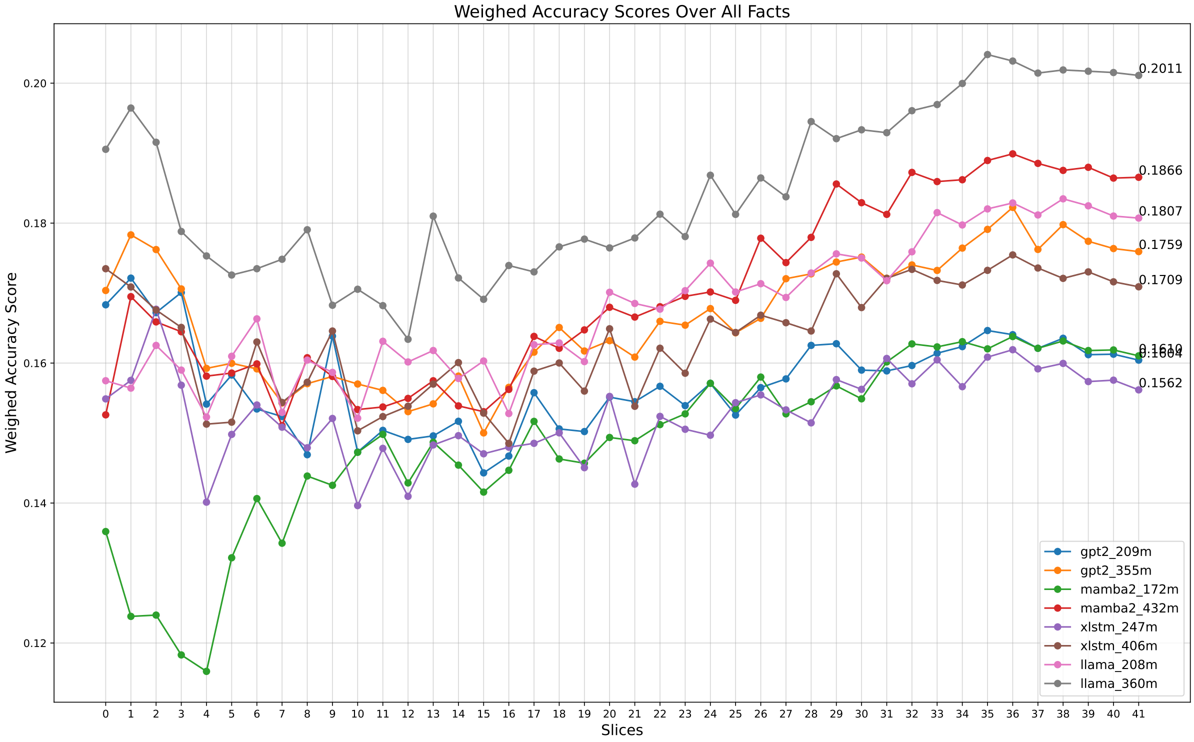


Figure 13: Development of the weighted accuracy (WASB) throughout the pre-training as measured on BEAR-big.

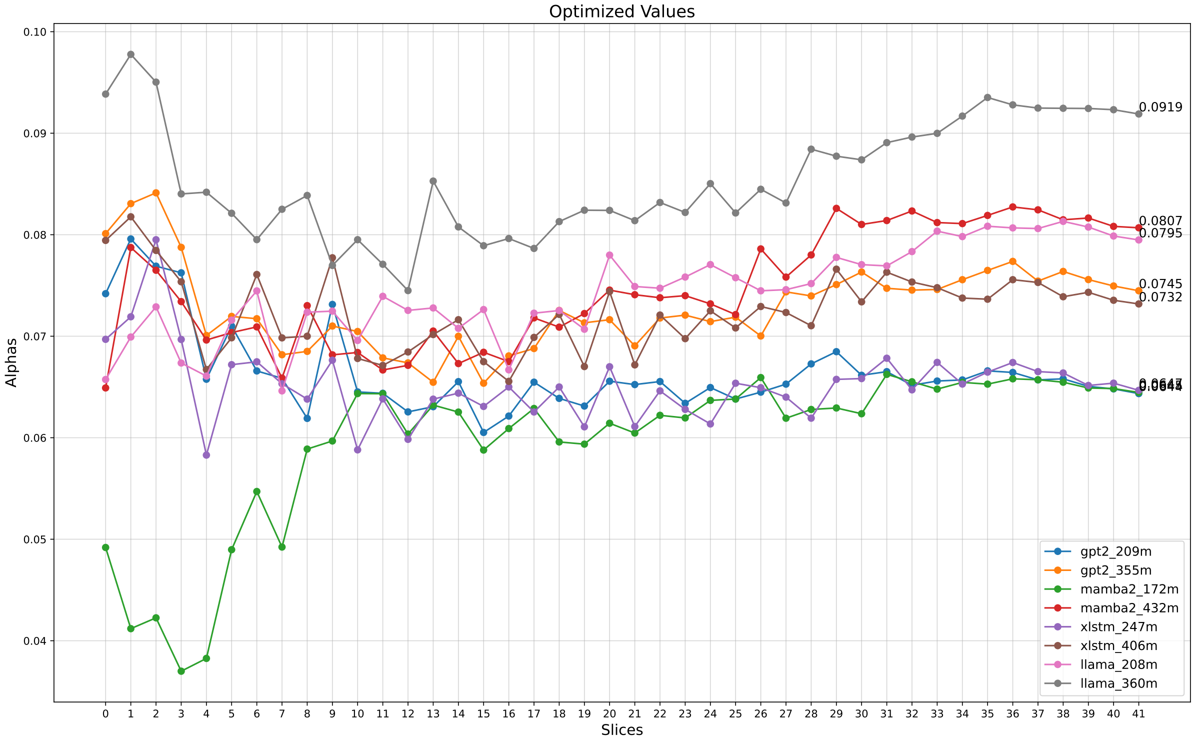


Figure 14: Development of α_m throughout the pre-training as measured on BEAR-big.

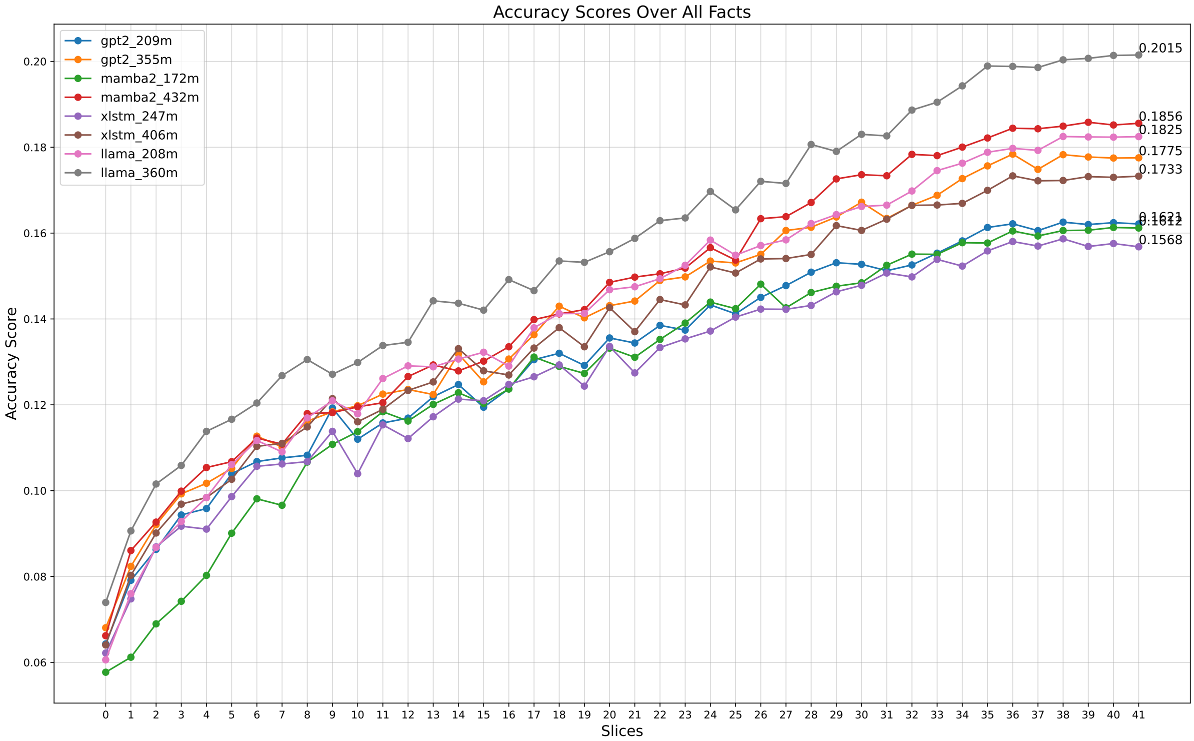
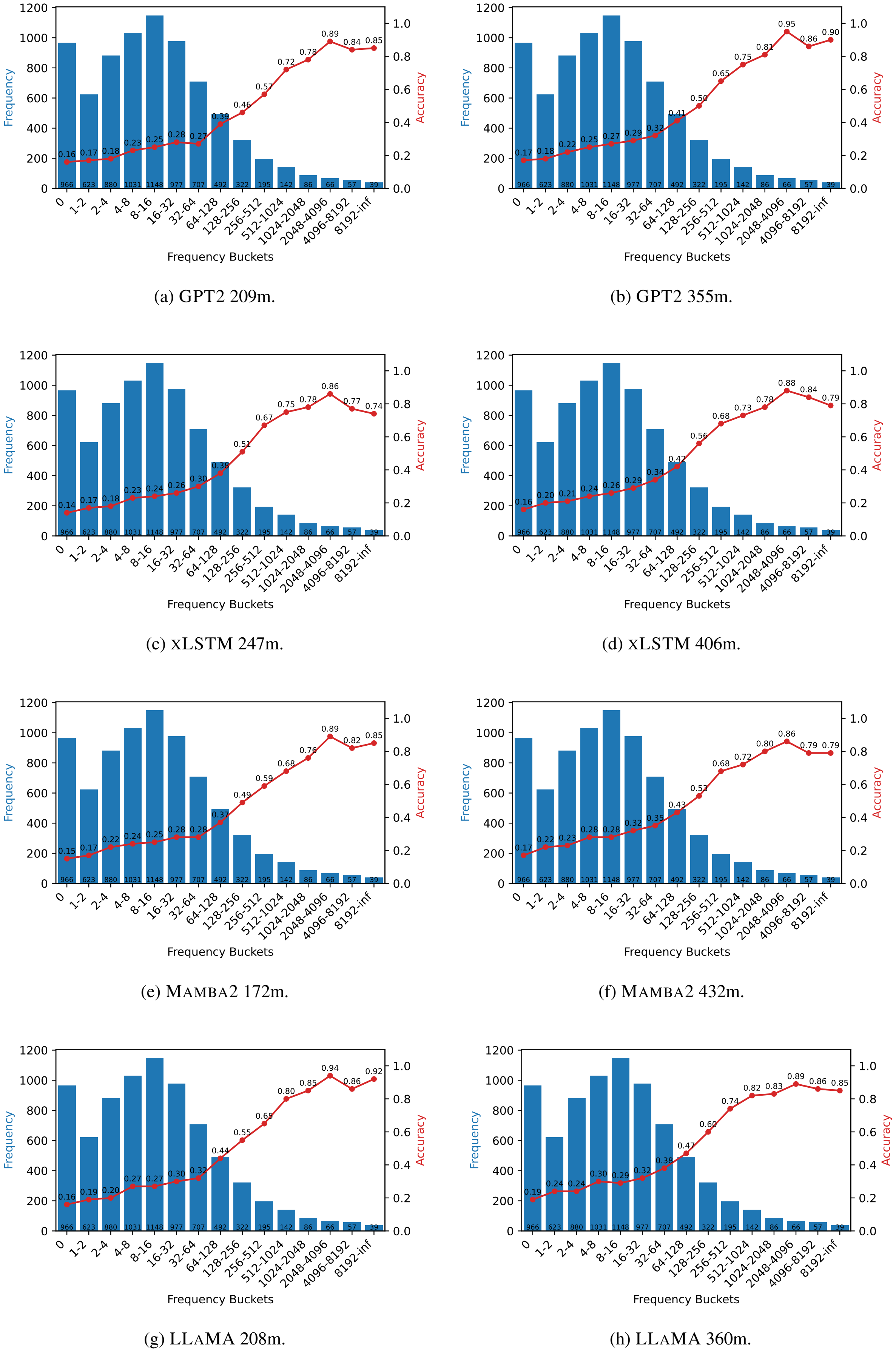


Figure 15: Development of the accuracy throughout the pre-training as measured on BEAR-big.



(a) GPT2 209m.

(b) GPT2 355m.

(c) xLSTM 247m.

(d) xLSTM 406m.

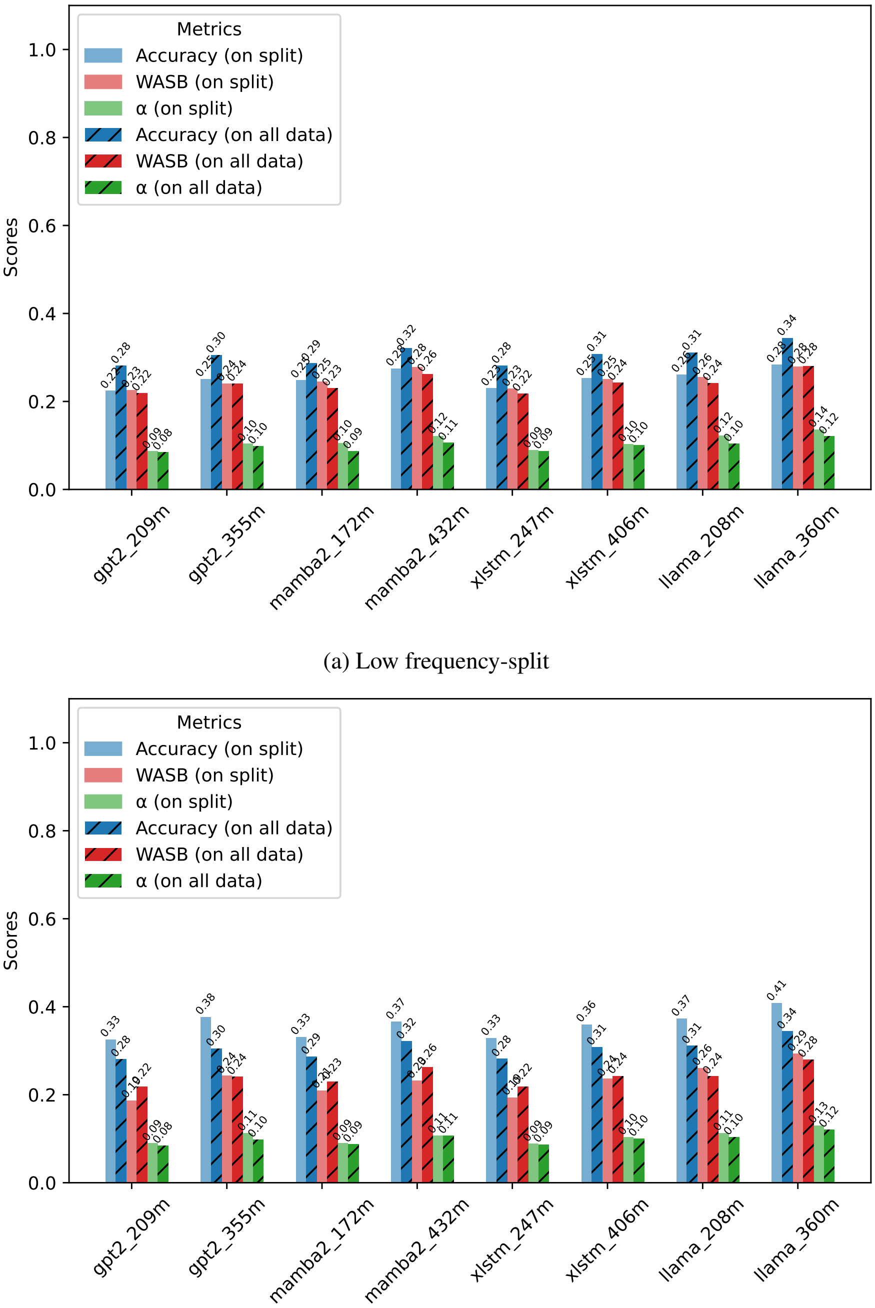
(e) MAMBA2 172m.

(f) MAMBA2 432m.

(g) LLAMA 208m.

(h) LLAMA 360m.

Figure 16: Frequency Bucket Accuracy of the model’s final state as measured on BEAR.



(a) Low frequency-split

(b) High frequency-split

Figure 17: Accuracy, WASB and α scores on the low and high frequency splits and entire data set for comparison on BEAR.