LLäMmlein 🐑: Compact and Competitive German-Only Language Models from Scratch

Jan Pfister🐑 and **Julia Wunderle**🐑 and **Andreas Hotho**
Data Science Chair
Center for Artificial Intelligence and Data Science (CAIDAS)
Julius-Maximilians-Universität Würzburg (JMU)
{lastname}@informatik.uni-wuerzburg.de

Abstract

We create two German-only decoder models, LLäMmlein 120M and 1B, transparently from scratch and publish them, along with the training data, for the German NLP research community to use[1]. The model training involved several key steps, including extensive data preprocessing, the creation of a custom German tokenizer, the training itself, as well as the evaluation of the final models on various benchmarks. Throughout the training process, multiple checkpoints were saved and analyzed using the SuperGLEBer benchmark to monitor the models' learning dynamics.

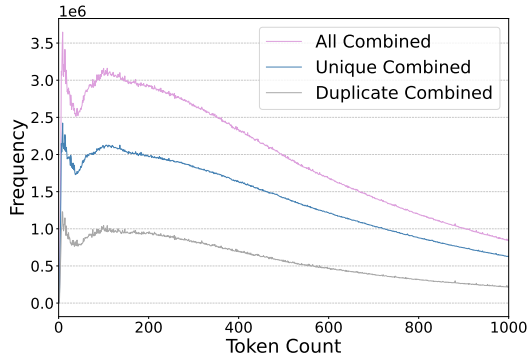
Compared to state-of-the-art models on the SuperGLEBer benchmark, both LLäMmlein models performed competitively, consistently matching or surpassing models with similar parameter sizes. The results show that the models' quality scales with size as expected, but performance improvements on some tasks plateaued early, offering valuable insights into resource allocation for future model development.

1 Introduction

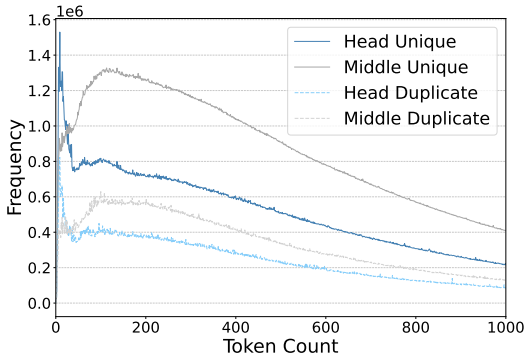
The recent success of Large Language Models (LLMs) has brought progress across many areas and languages. However, much of this progress has been concentrated on English, resulting in a notable gap for other languages, including German. In comparison to English, where large research labs and companies actively compete and heavily invest in training and tuning new models on new datasets commonly from scratch nearly every week, the German LLM landscape seems left behind. Previously, a few BERT or smaller GPT models have been pre-trained in German corpora (Chan et al., 2020; Scheible et al., 2020). However, as of recent developments, LLMs have become larger and have either been multilingually trained on other languages as well as German (e.g. mT5 (2021)), trained on English data and then finetuned to German (e.g. LeoLM (2023)) or there is no usable information about the (German) pretraining data at all (e.g. Mistral (2023)). This always makes non-English (i.e. German) language modeling more of an "afterthought" in these models, which can lead to various performance deteriorations on non-English languages (Anschütz et al., 2023; Virtanen et al., 2019). These issues can even be noticed in the most recent Llama 3 (Dubey et al., 2024) instruct models, which often fall back to answering in English, despite the previous conversation being held in German. These issues highlight the pressing need for dedicated, German-specific research and model development to identify and address the limitations of the current German LLM landscape. Specifically, there is a significant lack of transparency regarding the German-language data used to train existing models and how this data contributes to their capabilities. To address this gap, we (plan to) open-source our entire training pipeline, including data, code, and artifacts, to provide comprehensive insights into the training process, foster reproducibility, and encourage collaboration in this analysis within the research community.

To train the model, we have carried out the following steps: First, we filter and preprocess a large dataset, sourced from RedPajama V2 (Computer, 2023), to only contain high-quality German data. Second, we build a new German tokenizer with a vocabulary size of 32,000 tokens and train it on different amounts of data to compare its performance to existing German tokenizers. Next, we pretrain two exclusively German autoregressive language models from scratch on this dataset: *LLäMmlein* 120M and 1B. Training two models not only provides a basis for comparison, but also enables analysis of scaling effects on model performance and efficiency. In line with the Pythia experiments (Bi-

🐑 These authors contributed equally to this work.
[1] https://www.informatik.uni-wuerzburg.de/datascience/projects/nlp/llammlein/



(a) Token count distribution for the entire dataset (pink), the combination of all unique data (blue) and duplicate partition (gray).



(b) Token count distribution for each partition separately: head unique, middle Unique, head duplicate and middle duplicate

Figure 1: Redpajama statistics based on gbert-large tokenizer

derman et al., 2023), we keep track of and publish intermediate checkpoints throughout the whole process. This might e.g. help determine whether training could have been concluded earlier if additional data no longer yield meaningful improvements, optimizing both time and resources for future projects, but also enables us to analyze the learning dynamics of the model. Finally, we evaluate the models on a variety of tasks, including SuperGLEBer (Pfister and Hotho, 2024) and lm-evaluation-harness-de (Plüster; Gao et al., 2021) benchmarks, to assess their performance and compare them to existing models.

In summary, our main contributions are as follows: 1. Cleaning, filtering and preprocessing of the RedPajama dataset 2. Training new German tokenizer 3. Training of two German-only language models, LLäMmlein 120M and 1B completely from scratch 4. Evaluation of training process 5. Comparison to state of the art language models on German. 6. We publish our models, code and data to enable further research and development in the German LLM landscape.

2 Methodology

Pretraining and evaluating a German LLM from scratch, end-to-end involves several steps, including dataset preprocessing (section 2.1.2), tokenizer training (section 2.2), model pretraining (section 2.3), model evaluation using German downstream task and translated prompt-based few-shot QA tasks (section 2.4), and downstream adaptations (section 2.5).

2.1 Dataset

RedPajama V2 (Computer, 2023) is an open, multilingual dataset designed for training large language models. It consists of over 100 billion text documents collected from 84 CommonCrawl snapshots between 2014 and 2023 and encompasses multiple languages, including English, German, French, Italian, and Spanish. The dataset was originally preprocessed using the CCNet pipeline (Wenzek et al., 2020) leading to about 30 billion overall documents further enriched with quality signals and duplicate indicators. Using a perplexity score, the RedPajama dataset was divided into three quality categories: head, middle, and tail.

2.1.1 Dataset Analysis

The aim of the following preliminary dataset analysis is to gain a deeper understanding of the German portion of the dataset used. The "official" estimate of the size of the German segment within the RedPajamaV2 dataset, derived through extrapolation from a smaller sample analyzed with Mistral-7B, is approximately 3 trillion German tokens (Computer, 2023). We will perform an exploratory analysis of the dataset to gain a clearer understanding of the actual amount of German data it contains, alongside its domain distribution and the most prevalent data sources.

Statistics Our own analysis on the entire head and middle part, using the gbert-large tokenizer, led to a token count of 2.7 trillion German tokens for head and middle categories combined.

Figure 1a shows the data distribution of the full German dataset in pink, the full unique data including, head and middle, in blue and the partition of

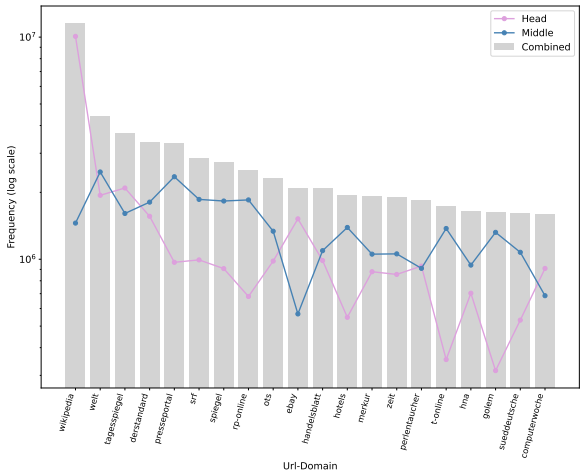


Figure 2: Top 20 most frequent domains across the full dataset in gray with frequencies in head and middle partitions separately.

duplicates in gray (according to the "is_duplicate" key). Most samples are unique (1.9 billion samples) and only significantly less are marked as duplicates (777 million samples) across the entire dataset. All three lines exhibit similar patterns. The most common token frequency per document can be found at nine, with approximately 3.6 million occurrences across the entire dataset. A second peak occurs at around 100 tokens per document. In total, the 2.7 trillion German tokens are distributed across samples with lengths ranging from 1 to 1,034,799 tokens, averaging approximately 1,000 tokens per sample.

Figure 1b breaks down the distribution of each partition, i.e. head unique, middle unique, head duplicate and middle duplicate separately. The middle unique partition contains the largest amount of data, with approximately 1.2 billion samples, which corresponds to 45% of the full dataset. The head unique partition, by comparison, includes around 400 million fewer samples.

Domain Analysis The dataset contains content crawled from various domain names. Figure 2 displays the top 20 sources from which the data was collected, with the overall count illustrated as gray bars and separate plots for the head (pink) and middle (blue) unique splits.

Wikipedia clearly stands out as the largest contributor, with a combined total of over 11.5 million samples. Among these, about 10 million entries belong to the head category, while about 1.45 million stem from the middle partition. This distribution aligns with the fact that the split into head, middle and tail was created using a perplexity score criterion based on a language model trained on Wikipedia (Computer, 2023) - consequently, texts closer in style to Wikipedia tend to be ranked higher. Besides Wikipedia, it is evident that news websites also constitute a significant portion of the dataset. For the middle split, "welt.de" emerges as the most frequent domain, contributing around 2.47 million samples. With the exception of domains like eBay, Handelsblatt, hotels and Perlentaucher, the list is largely dominated by general news outlets.

2.1.2 Further Dataset Preprocessing

To remove common web boilerplate, such as GDPR notices or similar repetitive content, we utilize a paragraph-level deduplication scheme powered by Dolma - a framework that enables efficient deduplication through a Rust-based Bloom filter (Soldaini et al., 2024). This ensures that highly redundant text is filtered out, improving the overall quality and diversity of the dataset. This approach may inadvertently over-remove valid and relevant content, such as short texts mistakenly treated as entire paragraphs and removed across the dataset. To mitigate this, and to preserve meaningful short text sections - such as lists or frequently occurring itemized phrases that are contextually significant - we excluded texts containing fewer than two words from the deduplication process. Additionally, we applied further cleaning to remove superfluous line breaks (\n) and whitespaces (\s).

Despite these efforts, some unusual artifacts, such as e.g. long sequences of guitar chords, remained. To address this, we built a token-to-word ratio filter. Here, we compared the word count (using whitespaces \s as separation) with the token count using the German GPT-2 tokenizer (Staatsbibliothek). According to our intuition, a large discrepancy between the two counts indicates abnormal or low-quality text, whereas a close match suggests valid content. A simple example illustrates this clearly: The phrase "Der Himmel ist blau" consists of 4 words and 4 tokens, so it is not removed by our filter. In contrast, "/de/c/trebic-unesco" counts as 1 word but 11 tokens, and should therefore be excluded by this token-to-word ratio filter.

Preliminary examinations and manual review suggested a ratio of tokens to words of eight as a valid threshold. Thus, paragraphs with ratios exceeding this threshold were excluded from the

dataset.

2.2 Tokenizer Training

We chose to train our own tokenizer due to the lack of an existing German tokenizer with a vocabulary size of 32,000. This decision ensures better alignment with the German language, the TinyLlama setup and dataset, while also enhancing transparency and traceability throughout the model's from-scratch training process. Using the setup outlined in TinyLlama (Zhang et al., 2024), we trained a Byte-Pair Encoding (BPE) tokenizer with a 32,000-token vocabulary tailored specifically for German.

2.3 Model Pretraining Framework

While there are several existing resources and repositories for training a Large Language Model (LLM) from scratch[2], we chose the TinyLlama GitHub repository as the backbone of our project (Zhang et al., 2024). It was used to pretrain the 1B English TinyLlama model from scratch before and builds upon the lit-gpt repository (AI, 2023), which provides robust tooling for data preparation, fine-tuning, pretraining, and deploying LLMs using PyTorch Lightning.

It includes all relevant features such as multi-GPU and multi-node distributed training with FSDP as well as flash attention 2. Additionally, it provides scripts to convert the models into HuggingFace format for easier use and distribution.

2.4 Model Evaluation Setup

2.4.1 Intermediate Checkpoint Evaluation

To get a better understanding of the training, we monitor the progress and regularly evaluate intermediate checkpoints on six representative SuperGLEBer tasks (Pfister and Hotho, 2024). These tasks were selected to include different task types to assess the performance of our model and will be discussed in the following:

- Classification:
  - Natural Language Inference (NLI) (Conneau et al., 2018): Predict if a hypothesis is entailed, neutral, or contradictory to a premise.
  - FactClaiming Comments (Risch et al., 2021): Binary classification of whether a text contains a fact-checkable claim.
  - DB Aspect (Wojatzki et al., 2017): Multi-label classification of categories and polarities in input sentences.
  - WebCAGe (Henrich et al., 2012): Predict if a word's sense matches across two contexts (true/false).
- Sequence Tagging
  - EuroParl (Faruqui and Padó, 2010): Token classification on European Parliament data.
- Sentence Similarity
  - Pawsx (Liang et al., 2020): Identify if sentence pairs are paraphrases by generating meaningful vector representations.

2.4.2 Final Model Evaluation

To assess general knowledge and abilities, we evaluated our final models on the full SuperGLEBer benchmark (29 tasks across classification, sequence tagging, question answering, and sentence similarity) (Pfister and Hotho, 2024), as well as machine-translated, prompt-based, few-shot QA tasks using the lm-evaluation-harness-de (if not stated otherwise, they are evaluated measuring accuracy and output length normalized accuracy (Plüster; Gao et al., 2021):

- ARC-Challenge-DE (Clark et al., 2018): Translated grade-school science questions (1,471 samples) from ARC-Challenge. Few-shot evaluation with 25 samples.
- MMLU-DE (Hendrycks et al., 2021): Translated version of MMLU with 6,829 multiple-choice questions across 57 topics (e.g., math, medicine, law). Few-shot evaluation with five samples.
- HellaSwag-DE (Zellers et al., 2019): Commonsense reasoning dataset with 11,000 translated samples, featuring incomplete sentences with multiple-choice completions. Few-shot evaluation with ten samples.
- TruthfulQA-DE (Lin et al., 2022): Dataset of 817 questions across 38 categories (e.g., health, law) designed to evaluate model truthfulness, particularly in handling misconceptions. Few-shot evaluation with zero samples. Performance is measured in single true (mc1) and multi true (mc2) (Plüster; Gao et al., 2021).

[2] a small subset: https://github.com/Hannibal046/Awesome-LLM#llm-training-frameworks

2.5 Downstream Adaptations

We fine-tune our model exemplarily to two downstream use cases: instruct-tuning and Bavarian. It should be noted that the results of these adaptations are not included in the evaluation of the model's performance on the SuperGLEBer benchmark, and we did not clean or preprocess the data for these tasks extensively before training. While this is not the main focus of this paper, we want to demonstrate the versatility of our model and its potential for further research and development.

2.5.1 Instruct-tuning

A common use case for modern LLMs is to generate text based on a given prompt. For this purpose, the model is tuned on input/output pairs, where the input is a user-given prompt and the output is the desired continuation. To this end, we employ LoRA (Hu et al., 2021) to fine-tune our model on various existing German instruct-tuning datasets.

2.5.2 Bavarian

To exemplarily adapt our model to a specific dialect, we fine-tune the model on a few thousand Bavarian samples using LoRA (Hu et al., 2021).

3 Experiments

3.1 Tokenizer

We follow the TinyLlama setup and use a Llama 2 Byte-Pair Encoding (BPE) tokenizer. We trained three tokenizers on different amounts of data and compared the results to two established German tokenizers, german-gpt2 and gbert-large in section 4.1.

- 1TB: Spans backward from the most recent data until 1TB of data is processed
- 2023-2021: Includes all splits of the high quality data split from years 2023 to 2021 (847GB)
- 2023_14: Consists of the most recent 2023_14 split (67GB)

3.2 Model Pretraining

For the training of our larger 1B model, we use the head and middle part of our preprocessed dataset, while for the smaller 120M model, we will use the head part only.

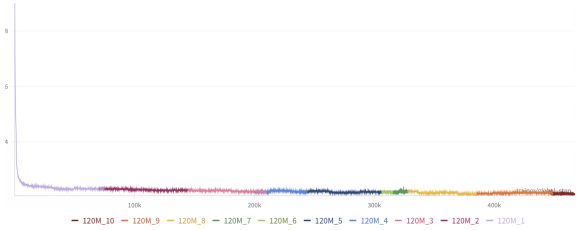


Figure 3: Loss curve of LLäMmlein 120M model. Each color indicates a run, resumed after a training interruption.

3.2.1 LLäMmlein 120M

Finally, we trained the LLäMmlein 120M model on the filtered head data (section 2.1), a maximum learning rate of 6e-4, grouped query attention of 4, a sequence length of 2048, a batch size of 1024, and the full-shard FSDP strategy. This model was trained on 32 L40 GPUs distributed across 16 nodes. This resulted in a total pretraining token count of: 954,485,440,512

Figure 3 displays the model's loss curve, where each training resume is distinguished by a unique color. Overall, 10 restarts were necessary: Due to cluster settings, training was resumed at least every two days, and additionally, training had to be restarted a few times to address GPU and NCCL errors.

3.3 LLäMmlein 1B

After the preliminary tests with the smaller model, we trained the 1B model. For training data, we utilized the full processed and filtered head and middle partitions, which resulted in a total token count of 2.7T (2,649,407,619,072) unique German tokens and 3T (3,000,001,101,824) German training tokens in total. As we closely follow the TinyLlama setup, we want to maintain the same step count, and we traverse our dataset once, and then again with a small overlap, like done in the original TinyLlama and Pythia setup. We trained our model on 64 A100 GPUs distributed across eight nodes. To keep the global batch size at 1024, we set the micro batch size to 16, and the maximum learning rate was 6e-4.

Initially, we attempted to estimate the runtime for replicating the original TinyLlama training run on our hardware. Based on our extrapolations, the process would have taken over 200 days using 16 A100 GPUs, compared to the 90 days reported for TinyLlama. By adapting the Fully Sharded Data Parallel (FSDP) strategy to a hybrid sharding ap-

Figure 4: Loss curve of LLäMmlein 1B model. Each color indicates a run, resumed after a training interruption.

proach, we reduced the extrapolated runtime to approximately 100 days, which we deemed satisfactory.

Next, we scaled our training to 64 GPUs, bringing the extrapolated runtime down to 36 days. Early in the training run, we identified inefficiencies related to improper use of the available InfiniBand. We halted the training, corrected the configuration, switched back to full sharding, and implemented dataset pre-caching in RAM on each node. After these optimizations, we restarted the training, achieving a final overall runtime of 32 days.

Figure 4 shows the loss curve of our 1B model. During pretraining, we had to resume training twice: once at 150,000 steps (blue) and at 590,000 steps (pink). The visible jumps in the loss curve correspond to different chunks of training data, each sampled from a distinct part of the dataset.

3.4 Downstream Adaptations

3.4.1 Instruct-tuning

We use LoRa (Hu et al., 2021) to fine-tune our model on various German instruct-tuning datasets, using supervised finetuning for three epochs with PEFT (Mangrulkar et al., 2022) and TRL (von Werra et al., 2020). We employ default hyperparameters and train for three epochs.

For training, we use the following datasets from huggingface: "LSX-UniWue/Guanako", "FreedomIntelligence/alpaca-gpt4-deutsch", "FreedomIntelligence/evol-instruct-deutsch", and "FreedomIntelligence/sharegpt-deutsch". We train and publish a separate adapter for each dataset, and also a combined model across all datasets.

3.4.2 Bavarian

To exemplarily adapt our model to a specific dialect, we fine-tune the model on about 25,700 Bavarian Wikipedia pages, using LoRA (Hu et al., 2021) with supervised finetuning, PEFT (Mangrulkar et al., 2022) and default hyperparameters. For this, we use the Bavarian column from the "cis-lmu/bavarian_to_english" dataset from HuggingFace.

4 Evaluation

4.1 Tokenizer

We tested the performance of our custom tokenizers on random samples and training splits, comparing the results to those of two established German tokenizers german-gpt2 (vocabulary size: 50,266) and gbert-large (vocabulary size: 31,102). While the different vocabulary sizes skew direct comparisons, the results still provide valuable insight into the relative performance of each tokenizer. The tables presented in Table 1 compare the token counts generated by the tokenizers when applied to two unseen data snapshots from the training dataset.

Notably, the german-gpt2 and gbert-large tokenizers obtain the lowest token counts. Interestingly, among our own tokenizers, the one trained on the smallest dataset (2023_14) performed better (i.e. produced fewer tokens) than those fitted on larger datasets. Our results suggest that the smaller dataset might allow the tokenizer to be more efficient by focusing on the most frequent tokens, while larger datasets might introduce more variation, leading to less efficient tokenization. Consequently, we chose to use the tokenizer trained on the 2023_14 snapshot.

4.2 Pretraining Process

During training, we regularly saved and evaluated checkpoints to monitor the training process (section 2.4.1).

4.2.1 LLäMmlein 120M

The results, assessed on six SuperGLEBER tasks - FactClaiming, EuroParl, Pawsx, NLI, DB Aspect and WebCAGe (see Section 2.4.1) - are summarized in Table 2. We compared LLäMmlein 120M's performance with models of similar parameter sizes, including german-gpt2, gbert-base and bert-base-german-cased. While our model consistently outperformed the only other decoder-only model gpt2, the BERT-based models showed superiority in the first three tasks (FactClaiming, EuroParl and Pawsx). This performance gap is consistent with the known limitations of autoregressive, decoder-only architectures on tasks such as sequence tagging and sentence similarity (Pfister and Hotho, 2024). LLäMmlein 120M particularly excelled in

Table 1: Comparison of our three created tokenizers with different training data sizes and other German tokenizers on two unseen training data samples: one from the head partition and one from the middle.

Tokenizer	Token Count
word count	80,782,685
german-gpt2	138,976,962
gbert-large	140,757,764
ours 1TB	183,720,038
ours 2023-2021	169,298,221
ours 2023_14	145,359,306

snapshot from middle

Tokenizer	Token Count
word count	46,509,357
german-gpt2	78,151,205
gbert-large	79,969,101
ours 1TB	105,481,995
ours 2023-2021	96,459,503
ours 2023_14	81,993,239

snapshot from head

Table 2: Results of different checkpoints of LLäMmlein 120M on six SuperGLEBer tasks compared to german_gpt2, gbert_base and bert-base-german-cased

Model	FactClaiming	EuroParl	Pawsx	NLI	DB Aspect	WebCAGe
010000	0.711	0.531	0.427	0.549	0.454	0.689
050000	0.717	0.536	0.428	0.549	0.452	0.688
100000	0.708	0.532	0.464	0.559	0.479	0.700
150000	0.702	0.516	0.474	0.575	0.474	0.692
200000	0.705	0.497	0.497	0.575	0.464	0.703
210000	0.715	0.493	0.489	0.578	0.475	0.685
250000	0.723	0.536	0.478	0.560	0.479	0.684
300000	0.712	0.525	0.497	0.615	0.498	0.682
350000	0.705	0.547	0.492	0.624	0.511	0.678
400000	0.713	0.522	0.488	0.627	0.511	**0.695**
450000	0.693	0.511	0.479	**0.638**	0.504	0.694
466509	0.711	0.538	0.489	0.629	**0.517**	0.687
german_gpt2	0.707	0.533	0.394	0.479	0.429	0.645
gbert_base	**0.751**	**0.616**	**0.561**	0.436	0.478	0.693
bert-base-german-cased	0.721	0.607	0.537	0.490	0.480	0.679

Table 3: Performance of LLäMmlein 1B across multiple training checkpoints on six SuperGLEBer tasks, with comparison to the best-performing models for each task in the benchmark.

Model	FactClaiming	EuroParl	Pawsx	NLI	DB Aspect	WebCAGe
010000	0.735	0.708	0.461	0.642	0.563	0.677
100000	0.734	0.662	0.511	0.709	0.607	0.699
190000	0.736	0.701	0.525	0.721	0.614	0.719
310000	0.744	0.656	0.521	0.725	0.611	0.720
400000	0.716	0.665	0.517	0.722	0.623	0.719
500000	0.733	0.712	0.539	0.734	0.613	0.720
600000	0.712	0.724	0.541	0.725	0.608	0.722
700000	0.737	0.676	0.529	0.727	0.630	0.722
800000	0.718	0.727	0.528	0.743	0.613	0.742
900000	0.732	0.718	0.542	0.748	0.634	0.733
950000	0.747	**0.732**	0.556	0.746	0.622	0.755
1000000	0.750	0.697	0.540	0.740	0.629	0.756
1100000	0.740	0.710	0.550	0.744	0.623	0.762
1200000	0.726	0.679	0.545	0.746	0.629	0.755
1300000	0.725	0.695	0.533	0.751	0.624	0.764
1350000	0.748	0.712	0.528	0.752	0.633	0.763
1400000	0.729	0.702	0.536	0.741	0.629	0.756
1420000	0.745	0.702	0.530	0.345	0.643	0.759
1430512	0.736	0.713	0.526	0.749	0.623	0.765
gbert_base	**0.751**	0.616	0.561	0.436	0.478	0.693
mbart_large_50	0.723	0.727	0.358	0.336	0.471	0.651
gbert_large	0.747	0.636	**0.654**	0.736	0.550	0.716
leo-mistral-7b	0.741	0.649	-	**0.807**	0.664	-
leo-hessian-7b	0.747	-	-	-	**0.669**	**0.781**

the more complex classification tasks. For the NLI task, it outperformed all other models starting from the first evaluated checkpoint "010000", with its best checkpoint exceeding the highest-performing bert-base-german-cased by 15%. Similarly, our model obtained the highest scores for DB Aspect and WebCAGE classification tasks.

Interestingly, performance improvements varied across tasks. We calculate the Spearman correlation coefficient r to measure the strength and direction of the monotonic relationship between training steps and performance, and the corresponding p-value to assess the statistical significance of the correlation. While for FactClaiming and EuroParl only minimal variation across checkpoints is observable, Pawsx ($r = 0.607$, $p = 0.04$), NLI ($r = 0.947$, $p < 0.0001$) and DB Aspect ($r = 0.909$, $p < 0.0001$) exhibited clear, linear improvement trends.

This suggests that while LLäMmlein quickly reached a plateau for certain tasks, i.e. those that might require more basic structure recognition (FactClaiming/EuroParl), it continued to learn and improve on some more complex tasks.

4.2.2 LLäMmlein 1B

We compared our results with those of all other models evaluated on the SuperGLEBer benchmark, focusing particularly on the best-performing model for each task (Table 3). Results of models that experienced out-of-memory (OOM) errors on an A100 80 GPU are indicated with a "-".

While LLäMmlein 1B may not secure the top results overall, it consistently ranks among the top three for each task listed in the table, demonstrating its reliability and stability compared to other models, which may exhibit significant performance variations. Interestingly, there is one exception in NLI at checkpoint 1,420,000, where performance drops by almost 40% compared to the previous checkpoint. However, performance recovers instantly at the following checkpoint. We are currently trying

Table 4: Performance comparison of LLäMmlein 120M to other language models on the lm-evaluation-harness-de including the four translated tasks: TruthfulQA, ARC-Challenge, HellaSwag and MMLU.

Model	TruthfulQA	ARC-Challenge	HellaSwag	MMLU
german gpt2	0.261|0.432	**0.195**|0.236	0.262|0.268	0.238|0.263
LLäMmlein 120M	0.247|0.404	0.194|**0.238**	**0.291**|**0.320**	**0.245**|**0.276**
LLäMmlein 120M Alpaka	**0.266**|**0.439**	0.178|0.235	0.269|0.277	0.231|0.268

to investigate the cause of this anomaly. Similar to the 120M model, LLäMmlein 1B is clearly outperformed for the sentence similarity task. However, it excels in the EuroParl task, setting a new maximum score with 0.732.

Examining task progress over time reveals noticeable improvements across all tasks, except for FactClaiming. Compared to the LLäMmlein 120M model, Spearman correlation analysis indicated a significant positive relationship between training time and performance of EuroParl (r = 0.431, p = 0.009) and WebCAGe (r = 0.92, $p < 0.0001$). All remaining tasks also demonstrated clear positive correlations with training time, suggesting that LLäMmlein 1B continues to benefit from extended training.

4.3 Final Model Evaluation

Detailed results for our models on SuperGLEBer can be found on the official SuperGLEBer website https://lsx-uniwue.github.io/SuperGLEBer-site/leaderboard_v1.

4.3.1 LLäMmlein 120M

After the evaluation of LLäMmlein's performance over time across different tasks, we now compare its effectiveness against other models across the whole SuperGLEBer benchmark. Figure 5a indicate that LLäMmlein significantly outperforms the german-gpt2 model, confirming its superiority among German decoder models in this size range. When comparing LLäMmlein to the two BERT models gbert-base and bert-german-cased, no statistically significant difference was found (Figures 5b and 5c). This is noteworthy, especially considering that BERT models generally excel in sequence tagging and similarity tasks. The comparable performance of LLäMmlein indicates that it competes well with established BERT-based models, despite BERT's inherent architectural advantages for some tasks.

We further evaluated our results on the lm-evaluation-harness-de evaluation benchmark for autoregressive models. Table 4 provides the normalized and unnormalized accuracy scores for ARC-Challenge, HellaSwag and MMLU and mc1 and mc2 for TruthfulQA (see Section 2.4.2). We compared LLäMmlein 120M with german-gpt2, the only other decoder model available at this parameter size. Additionally, we fine-tuned LLäMmlein on a German translated Alpaka dataset for instruction tuning. Notably, the instruction-tuned model achieved the best performance on TruthfulQA. For the other tasks, the base version of LLäMmlein demonstrated superior performance.

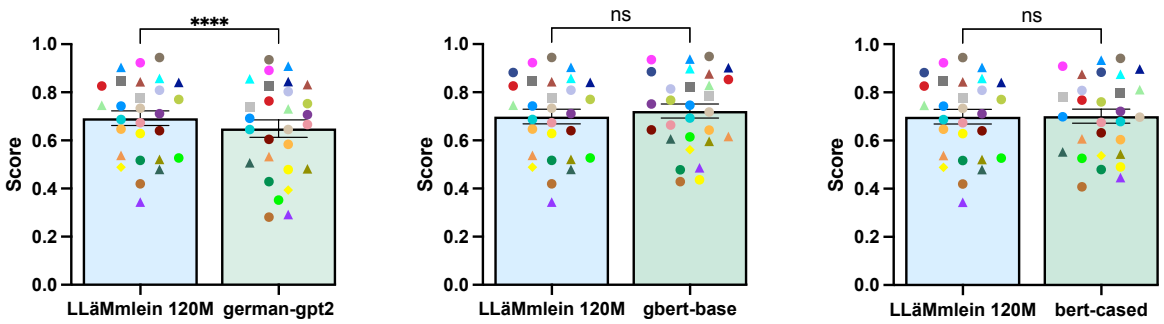
Table 5: Performance comparison of LLäMmlein 1B and its Instruction tuned variants as well as the similar sized Llama 3.2 1B and various larger models on the lm-evaluation-harness-de including the four tasks: TruthfulQA, ARC-Challenge, HellaSwag and MMLU.

Model	TruthfulQA	ARC-Challenge	HellaSwag	MMLU
Llama 2 7b	0.268|0.422	0.333|0.381	0.396|0.513	0.400|0.396
leo-hessianai-7b-chat	0.301|0.452	0.405|0.442	0.485|0.624	0.401|0.401
Llama 3 8B Ger	0.331|0.495	0.456|0.497	0.491|0.654	0.545|0.529
Llama 3 8B Ger Instruct	**0.364**|**0.530**	**0.506**|**0.538**	**0.515**|**0.664**	**0.559**|**0.555**
em_german_7b_v01	0.225|0.427	0.197|0.233	0.258|0.276	0.241|0.263
Llama 3.2 1B	**0.280**|0.407	0.265|0.310	0.339|0.412	0.284|0.302
Llama 3.2 1B Instruct	0.279|0.440	0.259|0.296	0.340|0.411	**0.343**|**0.343**
LLäMmlein 120M	0.247|0.404	0.194|0.238	0.291|0.320	0.245|0.276
LLäMmlein 1B	0.239|0.365	0.266|0.311	0.390|0.483	0.253|0.270
LLäMmlein 1B Guanako	0.244|0.375	**0.276**|**0.313**	**0.400**|**0.502**	0.258|0.274
LLäMmlein 1B Alpaka	0.244|**0.504**	0.229|0.249	0.253|0.254	0.228|0.250

4.3.2 LLäMmlein 1B

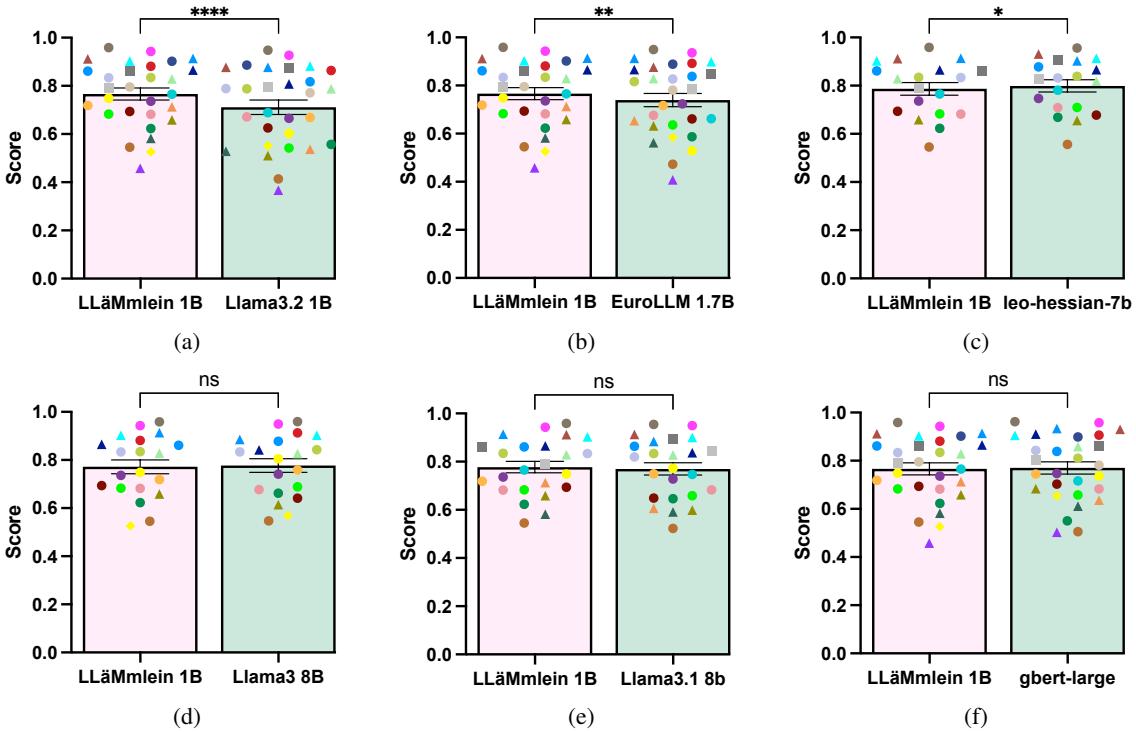
To further assess LLäMmlein 1B's performance, we conducted pairwise t-tests to compare results with other models on the SuperGLEBer benchmark. For consistency, we excluded tasks from the comparison, where a larger model lacked a score due to a cuda out of memory error. Among models with similar parameter size, we compared LLäMmlein 1B to Llama 3.2 with 1B parameters and EuroLLM with 1.7B parameters. Figures 6a and 6b show that our model is able to significantly outperform both similar-sized models. Leo-hessian-7B, with its larger architecture, showed superior performance, indicating that the size advantage of 7B models still holds in many cases (Figure 6c). Interestingly, no significant difference was found between LLäMmlein 1B and other much larger models, such as the German-finetuned Llama 8B (Figure 6d), Llama 3.1 8B (Figure 6e) as well as gbert-large (Figure 6f). This similarity with larger models highlights LLäMmlein 1B's efficiency and competitiveness.

Table 5 provides a comparative evaluation of the LLäMmlein 1B and its instruction-tuned variants, alongside similar-sized Llama 3.2 1B and larger models. Notably, the German finetuned Llama



(a) (b) (c)

Figure 5: Comparison of LLäMmlein 120M across the full SuperGLEBer benchmark with: (5a) german-gpt2, (5b) gbert-base and (5c) bert-base-german-cased. The asterisks indicate the level of statistical significance: "ns" denotes not significant ($p > 0.05$), while increasing significance is represented as follows: * ($p \leq 0.05$), ** ($p \leq 0.01$), *** ($p \leq 0.001$), and **** ($p \leq 0.0001$).



(a) (b) (c)

(d) (e) (f)

Figure 6: Performance comparison of LLäMmlein 1B accross the full SuperGLEBer benchmark with: (6a) Llama 3.2 1B, (6b) EuroLLM-1.7B, (6c) leo-hessian-7b, (6d) on German finetuned Llama 3 8B, (6e) Llama 3.1 8B and (6f) gbert-large. The asterisks indicate the level of statistical significance: "ns" denotes not significant ($p > 0.05$), while increasing significance is represented as follows: * ($p \leq 0.05$), ** ($p \leq 0.01$), *** ($p \leq 0.001$), and **** ($p \leq 0.0001$).

3 8B instruct model[3] achieved the highest scores across all tasks, demonstrating the advantage of increased model size and instruction tuning for this benchmark. Interestingly, the model has no significant performance gain compared to LLäMmlein 1B on the SuperGLEBer benchmark. This indicates that for actual generation tasks, size seems to matter more.

Regarding the smaller models, Llama 3.2 1B achieved the best unnormalized score for TruthfulQA. However, on the normalized score LLäMmlein 1B instruct-tuned on the alpaka dataset outperforms all 1B models by at least 6% as well as most of the larger models. In ARC-Challenge and HellaSwag, which are completion tasks requiring commonsense and contextual reasoning, the LLäMmlein 1B Guanako Instruct model consistently outperformed both the base LLäMmlein model and Llama 3.2 1B models. This result implies that the chat-tuned version of LLäMmlein 1B may have an advantage in generating coherent responses that align well with the completion task structure. Llama 3.2 1B Instruct achieved the best results for the general knowledge MMLU benchmark. Interestingly, for the LLäMmlein series, chat-tuning led to consistently higher scores across different tasks, whereas this trend did not appear for the Llama 3.2 1B models.

The results further highlight differences in task requirements: ARC-Challenge and HellaSwag, both completion tasks, differ structurally from TruthfulQA and MMLU, which are question-answering benchmarks. Smaller models, even when fine-tuned, may be less adept at true question-answering tasks that require a deep factual understanding. In addition, it is noticeable, that for our LLäMmlein models the normalized score is better than the unnormalized score most of the time, indicating that or model tends to generate rather long answers. When comparing the 120M version with the 1B model, the 1B consistently outperforms the smaller version by nearly 10%, except on the MMLU task—again highlighting the - expected - advantages of a larger model size.

5 Conclusion

We created two German-only decoder models, LLäMmlein 120M and 1B, from scratch. Achieving this involved preprocessing our training dataset in various steps, fitting a tokenizer to meet the specific requirements of our models and training, as well as evaluating the models.

We closely monitor the entire training process, evaluating several intermediate checkpoints to analyze the training dynamics. Our experiments revealed that the models learned different tasks at various speeds. Some tasks showed clear linearly correlated improvements over time, while others plateaued early on.

When compared to the state of the art on the SuperGLEBer benchmark, our models consistently matched or surpassed models of comparable sizes. In particular, LLäMmlein 1B outperformed the multilingual Llama 3.2 1B, underscoring the advantages of a monolingual model trained from scratch for language-specific tasks. Interestingly, mostly no significant difference was observed between our 1B model and larger models we evaluated, showcasing its effectiveness. However, for generative question answering, while our models performed similarly to other parameter-equivalent models, we observed notable differences to larger models, showcasing the clear advantage of the larger parameter counts for generation tasks and the limitations of smaller models in this regard.

Looking ahead, there are several avenues for future work: First, further analysis of the training process can be carried out, using the various intermediate checkpoints we saved and published. This opens the possibility for more granular evaluations of training dynamics. Furthermore, after our preliminary experiments with instruct-tuning, we want to point out the need for a high-quality and natively German instruct dataset, and plan to explore the potential of domain-specific fine-tuning. Domain adaptation and evaluation thereof could provide valuable insights into the model's capabilities and limitations.

Acknowledgments

The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project b185cb. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) – 440719683. Furthermore, we gratefully acknowledge the HPC re-

[3] https://huggingface.co/DiscoResearch/Llama3-DiscoLeo-Instruct-8B-v0.1

sources provided by the JuliaV2 cluster at the Universität Würzburg (JMU), which is partially funded by the German Research Foundation (DFG). The data science chair is part of the CAIDAS, the Center for Artificial Intelligence and Data Science, and is supported by the Bavarian High-Tech Agenda, which made this research possible.

References

Lightning AI. 2023. Lit-gpt.

Miriam Anschütz, Joshua Oehms, Thomas Wimmer, Bartłomiej Jezierski, and Georg Groh. 2023. Language models for German text simplification: Overcoming parallel data scarcity through style-specific pre-training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1147–1158, Toronto, Canada. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Together Computer. 2023. Redpajama: an open dataset for training large language models.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh,

Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Manaal Faruqui and Sebastian Padó. 2010. Training and evaluating a german named entity recognizer with semantic generalization. In *Conference on Natural Language Processing*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. A framework for few-shot language model evaluation.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Verena Henrich, Erhard Hinrichs, and Tatiana Vodolazova. 2012. WebCAGe – a web-harvested corpus annotated with GermaNet senses. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 387–396, Avignon, France. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego

de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Jan Pfister and Andreas Hotho. 2024. SuperGLEBer: German language understanding evaluation benchmark. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7904–7923, Mexico City, Mexico. Association for Computational Linguistics.

Björn Plüster. German benchmark datasets - translating popular llm benchmarks to german.

Björn Plüster. 2023. LeoLM: Igniting German-Language LLM Research. Accessed: 2024-11-15.

Julian Risch, Anke Stoll, Lena Wilms, and Michael Wiegand, editors. 2021. *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments*. Association for Computational Linguistics, Duesseldorf, Germany.

Raphael Scheible, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, and Martin Boeker. 2020. Gottbert: a pure german language model. *ArXiv*, abs/2012.02110.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. *arXiv preprint*.

Bayerische Staatsbibliothek. dbmdz/german-gpt2.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. *Preprint*, arXiv:1912.07076.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. Germeval 2017: Shared task on aspect-based sentiment in social media customer feedback.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024. Tinyllama: An open-source small language model. *Preprint*, arXiv:2401.02385.