Tokenizer Choice For LLM Training: Negligible or Crucial?

Anonymous ACL submission

Abstract

The recent success of Large Language Models (LLMs) has been predominantly driven by curating the training dataset composition, scaling of model architectures and dataset sizes and advancements in pretraining objectives, leaving tokenizer influence as a blind spot. Shedding light on this underexplored area, we conduct a comprehensive study on the influence of tokenizer choice on LLM downstream performance by training 24 mono- and multilingual LLMs at a 2.6 B parameter scale, ablating different tokenizer algorithms and parameterizations. Our studies highlight that the tokenizer choice can significantly impact the model's downstream performance and training costs. In particular, we find that the common tokenizer evaluation metrics *fertility* and *parity* are not always predictive of model downstream performance, rendering these metrics a questionable proxy for the model's downstream performance. Furthermore, we show that multilingual tokenizers trained on the five most frequent European languages require vocabulary size increases of factor three in comparison to English. While English-centric tokenizers have been applied to the training of multi-lingual LLMs in the past, we find that this approach results in a severe downstream performance degradation and additional training costs of up to 68%, due to an inefficient tokenization vocabulary.

1 Introduction

LLMs have shown impressive capabilities in many downstream tasks in a zero/few-shot setting such as summarization, reading comprehension, translation, and commonsense reasoning (Brown et al., 2020b; Touvron et al., 2023). To train a LLM, the currently established approach is to employ a tokenizer that splits the training documents into tokens where a token represents a word (Bengio et al., 2000), a sub-word (Schuster and Nakajima, 2012; Sennrich et al., 2015; Wang et al., 2020), or a single character (Gao et al., 2020b), and each token is represented in the model by an embedding vector that can be further processed.

The quality of a tokenizer can be assessed *intrinsically* and *extrinsically*. An intrinsic evaluation solely addresses the characteristics of tokenizers and their generated output in isolation, whereas the extrinsic evaluation measures the impact of the tokenizer on a downstream component, e.g., the Large Language Model (LLM).

While many different tokenization approaches have been proposed, ranging from character-based to word-based methods, the potential impact of different tokenizers is underexplored w.r.t. LLMs, especially in the context of multilingual LLMs. Recent work proposed by Petrov et al. (2023) demonstrates that carelessly designed tokenizers applied to the training of multilingual LLMs result in severe inequalities and limitations across languages. Text passages translated into different languages resulted in tokenized sequences that differ in length up to a factor of 15, affecting inference costs and latency during inference. Furthermore, it is known that the learning of long-range dependencies (Vaswani et al., 2017), is an essential property for effectively learning transformer-based LLMs. Given a fixed sequence length, learning to relate words far apart in the input text is impossible for languages whose text is excessively fragmented by the tokenizer.

Despite the importance of tokenizers and the potentially severe impact of poorly performing tokenizers, there exists no extensive study so far that holistically investigates the intrinsic and extrinsic tokenizer performance in a monolingual and multilingual setting with a focus on decoder-only models, which represent the backbone of current LLMs.

In this work, we address this gap and conduct an extensive study in which we measure the impact of the tokenizer on the model performance. In particular, we make the following contributions:

- We conduct a study investigating the intrinsic tokenizer performance.
- We conduct a study investigating the extrinsic tokenizer performance, i.e., the impact of the tokenizer on the model's downstream performance.
- We investigate whether a correlation between the intrinsic and the extrinsic tokenizer performance exists.

2 Related Work

This section provides an overview of tokenization algorithms and their usage in encoder- and decoder-only transformer models.

2.1 Tokenization Approaches

Word Tokenization. The most basic tokenization approach is the splitting of sequences based on white spaces and considering each word as a token (Bengio et al., 2000).

Subword tokenization. This class of algorithms subsumes all data-driven tokenization approaches which can decompose words into subwords/multiple tokens and currently represent the established tokenization approach upon which LLMs rely (Kudo and Richardson, 2018; Petrov et al., 2023). Because subword tokenizers decompose words into subwords, they can process out-of-vocabulary words by merging subwords from the vocabulary (Kudo and Richardson, 2018). Examples of popular subword tokenizers are WordPiece (Schuster and Nakajima, 2012), BPE (Gage, 1994; Sennrich et al., 2015), Byte-Level BPE (BBPE) (Wang et al., 2020), and Unigram (Kudo, 2018).

Character Tokenization. Tokenization can also be performed on a character level or based on UTF-8 bytes. However, this results in an increased sequence length, which becomes computationally expensive in the transformer architecture, the current predominated architecture for LLMs due to the quadratic complexity of the self-attention layer in the sequence length (Vaswani et al., 2017). Though, several approaches have been proposed to address this limitation (Gao et al., 2020b; Tay et al., 2021; Xue et al., 2022; Clark et al., 2022; Yu et al., 2023).

2.2 Tokenizers in Transformers Models

Tokenizers in Encoder Models Most research on tokenization has been conducted on encoder models. Rust et al. (2021) investigated whether the tokenizer choice impacts the downstream performance of multi- and monolingual BERT (Devlin et al., 2018) models. Zhang et al. (2022) showed that better machine translation performance is often obtained when languages are equally sampled during the tokenizer training. Toraman et al. (2023) trained several medium-sized language models for Turkish and suggested that different subword tokenizers perform roughly equivalent, whereas word- and character-level tokenizers perform drastically worse on downstream tasks. Finally, (Chirkova and Troshin, 2022) analyzed the effect of employing different tokenizations on code-related tasks and demonstrated that carefully configured tokenizers could reduce average sequence length up to 40% or allow for small downstream performance improvements by up to 2% at a lower compression rate.

Tokenizers in Decoder Models An overview of current mono- and multilingual LLMs is provided in (Lin et al., 2022; Shliazhko et al., 2022; Scao et al., 2022). Stollenwerk (2023) evaluated the intrinsic metrics of the GPT-SW3 (Ekgren et al., 2023) tokenizer that focused on the Nordic languages. As part of their work, Shliazhko et al. (2022) ablated different tokenizer pre-processing approaches while keeping the tokenizer algorithm, the vocabulary size, and the employed implementation fixed. In none of the other major LLM publications, the extrinsic tokenizer performance has been studied.

3 Approach

To investigate the tokenizer impact on the model performance, we conducted an extensive ablation study. In detail, we created dedicated datasets for the training of the tokenizers and the models, trained BPE and Unigram tokenizers, and for each tokenizer we trained decoder-only models with a size of 2.6B parameters while keeping the remaining configuration (i.e., dataset and model hyper-parameters) fixed. This allowed us to measure the tokenizer's impact on the model's downstream performance in isolation.

3.1 Data

While creating our tokenizer and model training datasets, we ensure that the mixture proportions of data domains (Wikipedia, books, web text) follow the same distribution to avoid a domain shift between tokenizers training and model training. We created *two datasets* with 70B words where one of the datasets is monolingual, containing English documents, and the second is a multilingual dataset comprised of English, German, French, Italian, and Spanish documents. Our datasets are filtered and deduplicated and consist of web-crawled data (80%) and curated data (20%), comparable to related datasets used to train LLMs. In the multilingual dataset, the amount of web-crawled data is equally distributed across languages in terms of number of words. Further details about our data pipeline and the data composition are described in Appendix A.

3.2 Tokenizer

Our studies rely on the two established tokenization algorithms, BPE and Unigram, and their implementation in the *Huggingface tokenizer* library (Moi and Patry, 2023) and the *SentencePiece* library (Kudo and Richardson, 2018). We considered both libraries in order to investigate the effect of differences in the pre-and post-processing steps and potential differences in the implementations. Due to missing pre-processing options for Huggingface's Unigram implementation, which causes a large discrepancy in the resulting vocabulary compared to SentencePiece's implementation of Unigram, we omitted the training of Unigram tokenizers based on Huggingface. Overall, we trained 24 different tokenizers, where one-half of the tokenizers were monolingual English tokenizers, and the other half of the tokenizers were multilingual tokenizers. Besides the tokenizer algorithm, language composition, and employed tokenizer library, we also varied the vocabulary size. Concrete tokenizer configurations are described in the Appendix B.

3.3 Models

To measure the impact of our trained tokenizers on the model downstream performance, we trained one model for each tokenizer. In particular, for each of our 24 trained tokenizers, we trained a 2.6B transformer-based decoder-only model on up to 52B tokens following the scaling law proposed by (Hoffmann et al., 2022a). Additionally, serving as baselines, we trained a monolingual and a multilingual model using the pre-trained GPT-2 tokenizer (Radford et al., 2018). All models have been trained based on the causal language modeling training objective.

3.4 Evaluation

To assess the impact of the tokenizers on the model downstream performance, we first performed an intrinsic tokenizer evaluation, followed by an extrinsic evaluation, and finally, we investigated whether a correlation between both evaluation approaches is given.

The intrinsic evaluation aims to assess the generated output of tokenizers based on *fertility* and *parity*. Furthermore, the tokenizer's vocabulary overlap with other tokenizers is computed. The intrinsic evaluation does not assess the impact of tokenizers on the model performance.

Fertility, the most common metric to evaluate a tokenizer's performance (Scao et al., 2022; Stollenwerk, 2023; Rust et al., 2021), is defined as the average number of tokens that are required to represent a word or document. For a tokenizer T and dataset A, the fertility can be calculated as the number of tokens in A (when T is applied) divided by the number of words in A. We calculate the fertility on a held-out set (10,000 documents), which was not used for the tokenizer training. For calculating the words of a document, we used whitespace splitting. Higher fertility scores correspond to weaker compression capabilities of the tokenizer.

Parity (Petrov et al., 2023), which has been recently proposed, assesses how fairly a tokenizer treats equivalent sentences in different languages. A tokenizer T achieves parity for language A with respect to language B if $\frac{|T(s_A)|}{|T(s_B)|} \approx 1$, where s_A and s_B denote the sets of all sentences in the corpora of languages A and B, respectively, and the ratio $\frac{|T(s_A)|}{|T(s_B)|}$ is defined as premium. We use the FLORES-200 (Goyal et al., 2022) parallel corpus, consisting of the same sentences human-translated into 200 languages. We calculate the parity values for each tokenizer and the four non-English languages with respect to English (see Fig. 2 for an overview).

The extrinsic evaluation aims to explicitly assess the impact of a tokenizer on the model's downstream performance. We selected a comprehensive set of downstream tasks (see Section 5.1) to measure the downstream performance.

Additionally, we computed the impact of a tokenizer on the average computational costs of a given model per word during training. The computational costs during training for one step including the forward and the backward pass can be estimated by

$$C = 96Bslh^2 \left(1 + \frac{s}{6h} + \frac{V}{16lh}\right), \qquad (1)$$

given a model with batch size B, sequence length s, l layers, hidden size h and vocabulary size V (Narayanan et al., 2021). The costs per token can be derived by $C_{\text{token}} = C/Bs$ and the average costs per word by $C_{\text{word}} = C_{\text{token}} \times$ fertility. The Results are discussed in Section 5.3.

4 Intrinsic Tokenizer Evaluation

In our intrinsic evaluation, we first compare the fertility and parity of the trained tokenizers (Section 4.1) and subsequently the overlap of their vocabularies (Section 4.2).

4.1 Fertility & Parity

Applying the described fertility and parity evaluation to the mono-/multilingual tokenizers, our analysis highlights the following two major aspects, as visualized in Fig. 1 and Fig. 2.

Firstly, it can be observed that applying a monolingual tokenizer to multilingual data results in significantly higher fertility and parity scores (see Fig. 1a and Fig. 2). While multilingual tokenizers have lower fertility than monolingual English tokenizers on all non-English documents by a large margin, they are only slightly worse on tokenizing English documents, as shown in Fig. 1b.

Secondly, with increasing vocabulary size, fertility and parity reduce in all cases, which can be explained by the tokenizer requiring fewer subword tokens when tokenizing text given a larger vocabulary. However, it can be observed that for monolingual English tokenizers, the fertility is less dependent on the vocabulary when tokenizing English documents, implying that 33k might be a sufficiently large vocabulary.

4.2 Vocabulary Overlap

To analyze the tokenizer similarity, we calculated the vocabulary overlap. Particularly, we assess Huggingface's and SentencePiece's BPE implementations, as depicted in Table 1.

The overlap is roughly constant across different vocabulary sizes, and the total overlap tends to be rather low, despite being the identical algorithm only implemented by two different libraries. Consequently, the tokenizers produce different tokenized sequences, possibly affecting model training and downstream performance. Investigating the underlying reasons, the low overlap might be attributed to different configuration and pre-processing options in these libraries. Due to the larger thesaurus in multilingual documents, the overlap for the multilingual tokenizer is lower than for the English tokenizers.

	33k	50k	82k	100k
English	0.77	0.76	0.74	0.74
Multilingual	0.62	0.62	0.62	0.61

Table 1: Vocabulary overlap between the HuggingFace and SentencePiece BPE tokenizer for different vocab sizes.

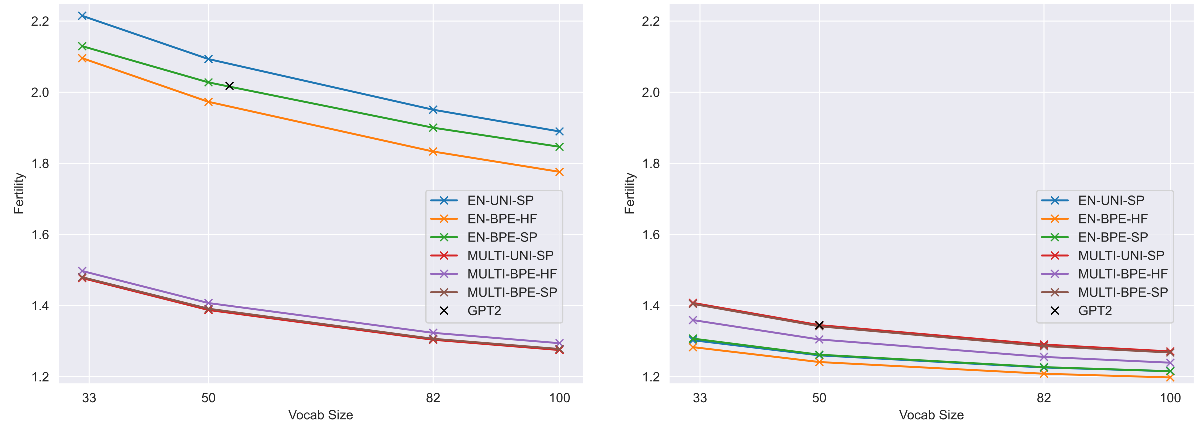
5 Extrinsic Tokenizer Evaluation

In the following, we describe the results of our extrinsic evaluation of tokenizers. Section 5.1 describes the experimental setup, Section 5.2 presents the downstream performance of the trained models based on the investigated tokenizers, and Section 5.3 analyzes the computational costs associated with each tokenizer when employed in a specific model.

5.1 Experimental Setup

To assess the impact of the tokenizers on the model downstream performance, we trained a decoder-only transformer model of size 2.6 B for each tokenizer. We trained our models for 52.6 B tokens following the scaling laws proposed by Hoffmann et al. (2022b), based on the causal language modeling training objective. The hyper-parameters are described in Table 10 in the Appendix C. We evaluated our models in zero-shot settings on a wide range of mono- and multilingual tasks:

- Natural language inference: XNLI (Conneau et al., 2018), MNLI (Williams et al., 2018), RTE (Wang et al., 2018), WNLI (Levesque et al., 2012), CB (De Marneffe et al., 2019)
- Question answering: X-CSQA (Goodman, 2001), XStoryCloze (Lin et al., 2022), PubMedQA (Jin et al., 2019)
- Reading comprehension: BoolQ (Clark et al., 2019)), LAMBADA (Paperno et al., 2016), RACE (Lai et al., 2017), MRPC (Dolan and Brockett, 2005).



(a) Non-English, multilingual documents

(b) English documents

Figure 1: Comparison of fertility scores between mono- and multilingual tokenizers applied to (a) Non-English, multilingual documents and (b) English documents.

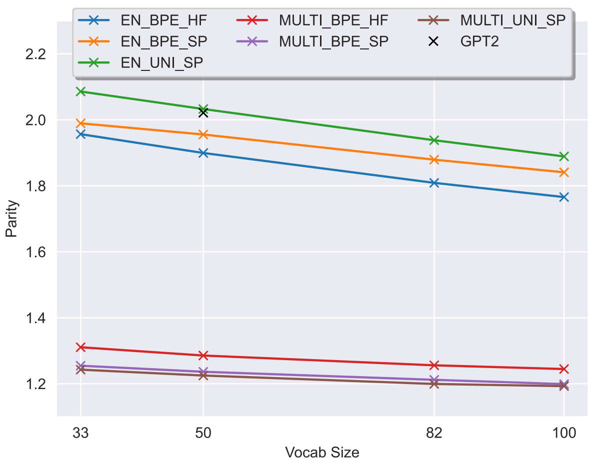


Figure 2: Comparison of parity scores between monolingual (English) tokenizer and multilingual tokenizers applied multi-lingual documents.

- Commonsense reasoning: HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), ARC (Clark et al., 2018), XCOPA (Ponti et al., 2020), XCDOAH (Goodman, 2001), WSC (Levesque et al., 2012), COPA (Roemmele et al., 2011)

- Classification: PAWS-X (Yang et al., 2019), GNAD10 (Schabus et al., 2017), SST (Socher et al., 2013), WIC (Pilehvar and Camacho-Collados, 2019), PIQA (Bisk et al., 2020)

Table 2 provides an overview of the number of tasks for each category and language.

Task	EN	DE	FR	ES	IT
NLI	6	1	1	1	0
QA	3	2	2	3	2
RC	3	1	1	1	1
CR	7	0	1	0	1
CL	3	1	0	1	0
	22	5	4	6	4

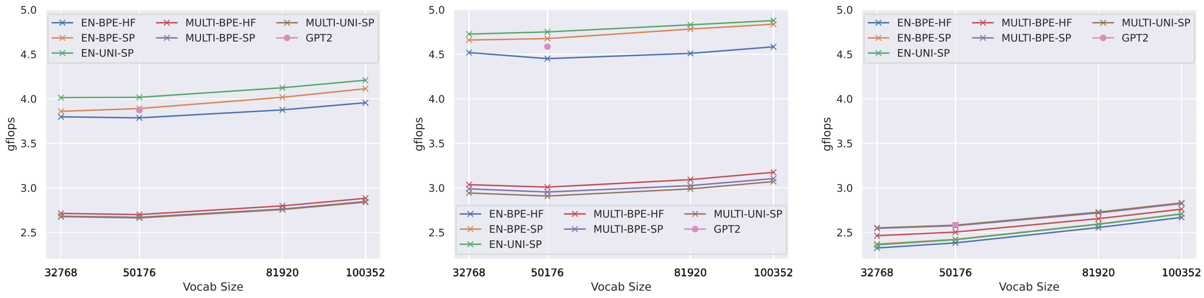
Table 2: Overview of the number of evaluation tasks for each language and the categories of Natural language inference (NLI), Reading comprehension (RC), Question answering (QA), Commonsense reasoning (CR) and Classification (CL).

5.2 Downstream Performance

We split our analysis of the downstream performance into several parts.

First, we discuss the overall results obtained for the investigated tokenizers, followed by presenting the impact of the tokenizer library (Section 5.2.1), the impact of the tokenizer algorithm (Section 5.2.2), and the impact of the vocabulary size (Section 5.2.3).

We present both, obtained results for selected single tasks (Table 3), and aggregated results across all tasks (Table 4). For the average performance across all tasks presented in Table 4, we computed weighted average to take into account the different number of tasks per language. In particular, we computed for each language the mean across all tasks, and then computed the mean over all language-means.



(a) Non-English documents (b) German documents (c) English documents

Figure 3: Average compute (GFLOPS) required to process a single word within (a) multilingual, (b) English, and (c) German documents within a full **training** pass (including the backward pass).

	Task	Min	Max	Rand.
EN	ARC-Easy	0.50	0.59	0.20
	HellaSwag	0.34	0.41	0.25
	MRPC	0.54	0.69	0.50
	PIQA	0.67	0.72	0.50
MULTI	XNLI FR	0.37	0.49	0.33
	XNLI EN	0.49	0.52	0.33
	X-CODAH ES	0.28	0.43	0.25
	10kGNAD	0.15	0.43	0.11

Table 3: Worst- and best-performing tokenizer for selected tasks and the random performance on this task.

Model	EN	MULTI
GPT-2-50	50.36	39.41
BPE-HF-33	49.13	40.52
BPE-HF-50	49.51	40.47
BPE-HF-82	48.71	40.24
BPE-HF-100	49.54	40.48
BPE-SP-33	**50.81**	40.28
BPE-SP-50	49.81	40.49
BPE-SP-82	48.99	41.21
BPE-SP-100	49.46	**41.44**
UNI-SP-33	50.28	40.30
UNI-SP-50	49.90	40.48
UNI-SP-82	49.65	41.20
UNI-SP-100	50.21	40.74

Table 4: Average accuracy of monolingual and multilingual tokenizers across all downstream tasks. Due to varying number of tasks per language, multi-lingual accuracies have been adjusted to each language contributing equally to the average.

Monolingual Tokenizer Table 4 demonstrates that the BPE-SP-33 tokenizer, on average, is the best-performing tokenizer, followed by the GPT-2 tokenizer. Interestingly, SentencePiece's implementation of BPE with a vocabulary size of 33k has been used for LLaMA2 (Touvron et al., 2023). Aggregated metrics provide a reasonable overview of the overall performance. However, it does not express potentially large performance differences across tasks. Therefore, we listed in Table 3 the obtained results for a list of selected tasks obtained by the best and worst performing tokenizer on this task. The results illustrate that the performance difference can be huge. For instance, for ARC-Easy, a commonsense reasoning task, the gap between the best and worst tokenizer is 9%.

Multilingual Tokenizer Table 4 shows that the BPE-SP-100 tokenizer is the best-performing tokenizer followed by the BPE-SP-82 tokenizer. Furthermore, Table 4 demonstrates that the GPT-2 tokenizer performs poorly, implying that using a pre-trained GPT-2 tokenizer to pre-train and fine-tune multilingual models should be **omitted**. The analysis of selected tasks (3) reveals that for multilingual tokenizers, the performance difference between tasks can be huge.

5.2.1 Impact of the Tokenizer Library

Table 5 demonstrates that BPE-SP, on average, outperforms BPE-HF in the monolingual and multilingual setting across all languages. The performance differences might be attributed to the differences in implementation details of the tokenizers' pre-and postprocessing, which could affect the vocabulary creation (see Section 4.2) and, consequently, the downstream performance.

	MULTI						MONO
Vocabulary	DE	FR	IT	ES	EN	AVG	EN
33	**36.75**	36.66	39.30	41.76	47.37	40.37	49.55
50	36.12	37.07	38.94	42.22	46.71	40.21	**49.90**
82	36.50	37.83	39.97	42.30	**47.80**	40.88	49.12
100	35.92	**38.07**	**40.13**	**42.64**	47.67	**40.89**	49.74
Algorithm and Library	DE	FR	IT	ES	EN	AVG	EN
BPE-HF	35.69	37.31	39.37	42.28	47.48	40.43	48.98
BPE-SP	**37.13**	37.45	**40.04**	41.96	**47.68**	**40.85**	49.77
UNI-SP	36.51	**37.66**	39.57	**42.56**	47.10	40.68	**50.01**

Table 5: Impact of the vocabulary size (upper), and tokenizer algorithm and library (lower), on the downstream performance. The accuracy scores are either averaged over the libraries and tokenizer algorithms (upper) or the different vocabulary sizes (lower).

5.2.2 Impact of the Tokenizer Algorithm

Furthermore, Table 5 shows that depending on the language, either the BPE or Unigram exhibits better performance. It is noteworthy that the Germanic languages German and English benefit from the BPE algorithm, whereas the Romanic languages French and Spanish benefited from Unigram. The experiments for Italian, a Romanic language as well, show a different pattern than the other two Romanic languages.

5.2.3 Impact of the Tokenizer Vocabulary

Analyzing the impact of the vocabulary size revealed that in the monolingual English setting, the smaller/medium-sized, i.e., a vocabulary size of 33k/50k performs better (Table 5) whereas in the multilingual setting, in all cases except for German, larger vocabulary sizes result in better downstream performance. Taking into account the results presented in Table 4 showing that in the monolingual English setting, the best-performing tokenizer on average across all tasks had a vocabulary size of 33k and that the best-performing multilingual tokenizer had a vocabulary size of 100k additionally supports the observation that for the monolingual English setting a small vocabulary size is beneficial and for the multilingual setting a large vocabulary size is required.

5.3 Computational Costs

Given a fixed model, the computational costs depend on the vocabulary size and the fertility of the tokenizer, as defined in Eq. (1).

While larger vocabulary sizes introduce additional computational costs, they might also result in lower fertility scores and, therefore, lower overall computational costs for processing a set of documents, as discussed in Section 4. However, our findings in Fig. 3 show that increasing the vocabulary size from 50k to larger vocabulary sizes increases the computational costs in all cases. This highlights that the potentially lower fertility of larger vocabulary sizes cannot compensate for the additional costs introduced by the larger vocabulary size.

Furthermore, we observe that the computational training costs for multilingual documents are significantly lower for multilingual tokenizers than for monolingual English tokenizers (Fig. 3a). In fact, Fig. 3b and Table 11 in the appendix demonstrate that the training costs can increase up to 68% (comparing Multi-UNI-SP-50 to EN-UNI-SP-100 for German documents) for a given dataset. Assuming that during training it is required to process a fixed set of documents (e.g., Wikipedia to learn specific facts) entirely and not only a given number of tokens, the choice of the tokenizer can significantly impact the computational costs for training on this corpus.

While we could observe large cost differences between multilingual and monolingual English tokenizers in the monolingual English setting, the difference in computational costs between multilingual and monolingual English tokenizers for processing English documents is marginal (Fig. 3c).

6 Correlation Between Intrinsic And Extrinsic Tokenizer Performance

This section investigates a possible predictive relationship of intrinsic tokenizer metrics (fertility and

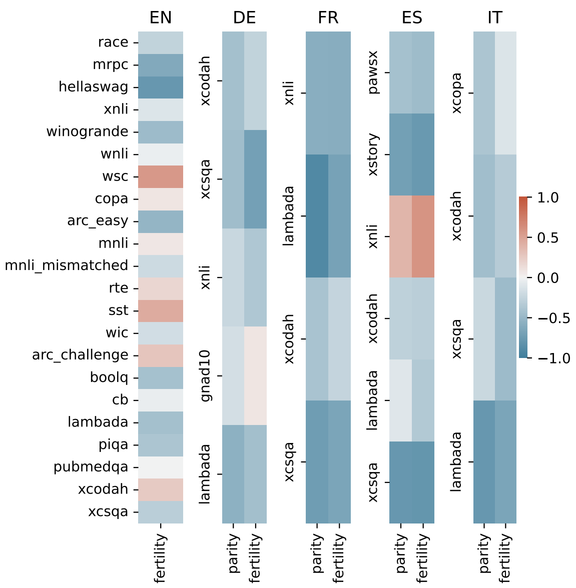


Figure 4: Spearman correlation of fertility/parity scores and downstream task performance for all five languages. We evaluated monolingual models on English tasks (left), whereas our multilingual models are evaluated across all non-English tasks. Pearson and Kendall correlation metrics showed a very similar picture.

parity) to the extrinsic model downstream performance.

As highlighted in the correlation heatmaps in Fig. 4, we find that there is no distinct correlation across all tasks and languages, demanding a more granular analysis. While for non-English tasks, we mainly observe a correlation between low fertility and higher downstream performance, the non-English tasks yield seemingly random positive and negative correlations. However, it should be noted that the number of multilingual tasks per language is much lower than for English and that for several multilingual tasks such as XSQA and LAMBADA, a similar correlation behaviour between the English tasks and their translated version can be observed.

Taking the fertility trends with varying vocabulary sizes (see Fig. 1) into consideration, we hypothesize that fertility only correlates with downstream performance in certain language-specific vocabulary size limits. For the English language, the tokenizers already provide low, close-to-convergence fertility scores for vocabulary sizes of 33k tokens. While additional tokens yield only minute fertility improvements, we presume that they do not capture morphological segmentations and, thus, can harm downstream performance and significantly increase the computation costs (see Section 5.3) in the end.

In contrast, for multilingual tokenizers, we observe significant fertility improvements with increasing vocabulary sizes. Due to the larger thesaurus induced by the additional languages, the tokenizer requires a larger vocabulary to allow a model to perform convincingly on all languages. Therefore, only within the non-convergence vocabulary range, we achieve a strong, negative correlation between fertility and downstream performance with varying vocabulary sizes.

In conclusion, intrinsic tokenizer metrics such as fertility and parity need to be taken with a grain of salt and supposedly are only predictive of downstream model performance in certain bounds. Low fertility scores might be regarded as a necessary criterion but not as a sufficient one.

7 Conclusion & Future Work

This work represents a fundamental step to a better understanding of the impact of the tokenizer on the models' downstream performance. We have shown that training tokenizers with a balanced share across languages achieve comparable low fertility and parity scores across all languages, which has important implications. Higher fertility results in up to 68% more computational costs during training and prevents the model from learning long-range dependencies in limited context windows.

Furthermore, we highlight that the tokenizer choice can significantly impact the model's downstream performance. We could show that the BPE algorithm applies well to mono- and multilingual settings. For English, we show that a vocabulary size of 33k is sufficient, whereas multilingual models based on our five considered languages require a up to three times larger vocabulary size. Moreover, we could show that the SentencePiece library outperforms the Huggingface tokenizer library.

Finally, we could demonstrate that there is no clear correlation between intrinsic and extrinsic tokenizer performance, but the correlation is rather task-specific. A small fertility value might be a necessary condition for good downstream performance but not a sufficient one.

In the future, we aim to investigate tokenizers for a larger set of languages, including very diverse languages, and investigate the impact of alternative tokenization approaches such as SAGE (Yehezkel and Pinter, 2023) that focus on context information during tokenizer training.

8 Limitations

Despite the extensiveness of our work, it faces the following limitations.

Firstly, we did not perform hyper-parameter optimizations for each tokenizer. This was a deliberate choice to avoid additional computational costs, considering that training all 26 models only once required ≈ 59.000 GPU hours.

Secondly, we did not investigate the effect of different random seeds on the model performance for a given tokenizer due to the additional computational costs. However, our results lay the foundation for future works that can further investigate the robustness of selected experiments.

Third, we did not investigate whether the results obtained could be extrapolated to larger model sizes, which we leave to future works. However, our finding that the BPE-SP-33 tokenizer is the best-performing tokenizer for the monolingual setting and the fact that this tokenizer has been used for training state-of-the-art models up to 65B (Touvron et al.) might indicate that our results also transfer to larger model sizes.

Finally, we did not provide results for a few-show setting since the metric of interest in the context of this work was the zero-shot downstream performance. Because we wanted to investigate whether the tokenizer choice impacts the model's downstream performance, we argue that restricting on one of the widely applied metrics, i.e., the zero-shot setting, is sufficient to answer this research question. One further advantage of focusing on the zero-shot scenario is that we do not introduce an additional variable represented by the choice of the few-shot examples. However, we encourage future works to investigate whether our results translate into the few-shot evaluation setting.

9 Ethical And Broader Impact

LLMs represent a disruptive technology that has received significant attention from the public and is widely used across societies speaking different languages. Therefore, ensuring a democratization of the technology across people of different languages will represent an important value. Our study highlights that neglecting multilingualism while training a tokenizer representing a core component required for training LLMs can cause severe disadvantages, such as increased training costs and decreased downstream performance, raising major ethical concerns. Furthermore, the increased training costs translate into an increased carbon footprint, which has an environmental impact. Our findings support an improved development and usage of this fundamental technology.

References

Julien Abadji, Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2021. Ungoliant: An optimized pipeline for the generation of a very large-scale multilingual web corpus. In Harald Lüngen, Marc Kupietz, Piotr Bański, Adrien Barbaresi, Simon Clematide, and Ines Pisetta, editors, *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-9) 2021. Limerick, 12 July 2021 (Online-Event)*, pages 1 – 9. Leibniz-Institut für Deutsche Sprache, Mannheim.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *Advances in neural information processing systems*, 13.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Nadezhda Chirkova and Sergey Troshin. 2022. CodeBPE: Investigating subtokenization options for large language model pretraining on source code. In *Deep Learning for Code Workshop*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL-HLT (1)*, pages 2924–2936. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

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. *CoRR*, abs/1803.05457.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: evaluating cross-lingual sentence representations. In *EMNLP*, pages 2475–2485. Association for Computational Linguistics.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Ariel Ekgren, Amaru Cuba Gyllensten, Felix Stollenwerk, Joey Öhman, Tim Isbister, Evangelia Gogoulou, Fredrik Carlsson, Alice Heiman, Judit Casademont, and Magnus Sahlgren. 2023. Gpt-sw3: An autoregressive language model for the nordic languages.

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

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020a. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Yingqiang Gao, Nikola I. Nikolov, Yuhuang Hu, and Richard H.R. Hahnloser. 2020b. Character-level translation with self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1591–1604, Online. Association for Computational Linguistics.

Joshua Goodman. 2001. A bit of progress in language modeling. *CoRR*, cs.CL/0108005v1.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Johannes Graën, Tannon Kew, Anastassia Shaitarova, and Martin Volk. 2019. Modelling large parallel corpora: The zurich parallel corpus collection. In *Proceedings of the 7th Workshop on Challenges in the Management of Large Corpora (CMLC)*, pages 1–8. Leibniz-Institut für Deutsche Sprache.

J. Graën, D. Batinic, and M. Volk. 2014. Cleaning the Europarl corpus for linguistic applications. In *Konvens 2014*. Stiftung Universität Hildesheim.

Najeh Hajlaoui, David Kolovratnik, Jaakko Vaeyrynen, Ralf Steinberger, and Dániel Varga. 2014. DCEP - Digital corpus of the European parliament. In *Proc. LREC 2014 (Language Resources and Evaluation Conference). Reykjavik, Iceland*, pages 3164–3171.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. 2022a. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pages 30016–30030. Curran Associates, Inc.

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, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022b. Training compute-optimal large language models. *CoRR*, abs/2203.15556.

Stefan Höfler and Michael Piotrowski. 2011. Building corpora for the philological study of Swiss legal texts. *Journal for Language Technology and Computational Linguistics*, 26(2):77–89.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. 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 2567–2577, Hong Kong, China. Association for Computational Linguistics.

P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation*

Summit, volume 5, pages 79—86. Asia-Pacific Association for Machine Translation (AAMT).

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

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *EMNLP 2018*, page 66.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC-2016)*.

Anthony Moi and Nicolas Patry. 2023. HuggingFace's Tokenizers.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, New York, NY, USA. Association for Computing Machinery.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *ACL (1)*. The Association for Computer Linguistics.

Aleksandar Petrov, Emanuele La Malfa, Philip HS Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. *arXiv preprint arXiv:2305.15425*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI*, pages 8732–8740. AAAI Press.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One million posts: A data set of german online discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1241–1244, Tokyo, Japan.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. *arXiv preprint arXiv:2204.07580*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Felix Stollenwerk. 2023. Training and evaluation of a multilingual tokenizer for gpt-sw3.

Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2021. Charformer: Fast character transformers via gradient-based subword tokenization. In *International Conference on Learning Representations*.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Sahinuc, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for turkish. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: open and efficient foundation language models, 2023. *URL https://arxiv.org/abs/2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9154–9160.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. 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 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Shaked Yehezkel and Yuval Pinter. 2023. Incorporating context into subword vocabularies. In *EACL*, pages 623–635. Association for Computational Linguistics.

Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. Megabyte: Predicting million-byte sequences with multiscale transformers. *arXiv preprint arXiv:2305.07185*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *ACL (1)*, pages 4791–4800. Association for Computational Linguistics.

Shiyue Zhang, Vishrav Chaudhary, Naman Goyal, James Cross, Guillaume Wenzek, Mohit Bansal, and Francisco Guzman. 2022. How robust is neural machine translation to language imbalance in multilingual tokenizer training? In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 97–116, Orlando, USA. Association for Machine Translation in the Americas.

Name	Language	#Words
Oscar	DE	11.200.000.000
Oscar	ES	11.200.000.000
Oscar	EN	11.200.000.000
Oscar	IT	11.200.000.000
Oscar	FR	11.200.000.000
Pile	DE	13.838.432
Pile	ES	21.990.512
Pile	EN	4.334.313.669
Pile	IT	7.946.402
Pile	FR	15.857.811
RedPajama	DE	143.907.461
RedPajama	ES	112.950.000
RedPajama	EN	4.663.646.781
RedPajama	IT	137.802.711
RedPajama	FR	139.749.147
RedPajama	Code	2.052.228.788
Misc	DE	600.844.912
Misc	ES	186.934.269
Misc	EN	1.337.030.904
Misc	IT	19.810.753
Misc	FR	211.147.445
Total		70.000.000.000

Table 6: Overview of the multilingual 70B words dataset with language, number of sampled words

A Corpora

Our web documents in the corpora consist of Oscars[1] (Abadji et al., 2021), that were generated by the ungoliant pipeline[2] based on three Common Crawl WET Archives (2022-27, 2022-49 and 2023-14).

The curated datasets consist of *The Pile* (Gao et al., 2020a), *RedPajama* (Computer, 2023), and single datasets that do not belong to a collection. From the Pile subcorpora, we selected: Phil Archive, PMC Abstracts, PMC Extracts, OpenWebText, NIH Exporterm, and Free Law Opinions V2. From RedPajama we use: ArXiv, Books, Github, StackExchange, and Wikipedia.

The remaining datasets are:

1. All the News V2.0[3] is a corpus of newspaper articles crawled from over 26 different publications from January 2016 to April 1, 2020.

2. Bundestag - Plenarprotokolle[4] comprises transcripts of sessions of the German Bundestag.

3. Bundesgerichtshof - Entscheidungen[5] is a collection of decisions of the German Federal Court.

4. CoStEP[6] is a cleaned-up and corrected version of the EuroParl corpus(Graën et al., 2014). (Koehn, 2005)

5. DCEP[7] is a companion corpus to CoStEP, containing documents published by the European Parliament. (Hajlaoui et al., 2014)

6. DNB Dissertations[8] is a collection of dissertations from the Deutsche Nationalbibliothek.

7. MAREC/IREC[9]: The MAtrixware REsearch Collection / The Information retrieval facility Research Collection is a patent corpus of over 19 million documents from the EP, WO, US, and JP patent offices.

8. Medi-Notice[10] is part of the Zurich Parallel Corpus Collection. It is a multilingual corpus compiled from information leaflets for

[1] https://oscar-project.org/
[2] https://github.com/oscar-project/ungoliant
[3] https://metatext.io/datasets/all-the-news-2.0

Name	Language	#Words
Oscar	EN	56.000.000.000
Pile	EN	4.893.724.288
RedPajama	EN	5.308.974.750
RedPajama	Code	2.299.301.635
Misc	EN	1.497.999.327
Total		70.000.000.000

Table 7: Overview of the English 70B words dataset with language, number of sampled words

[4] https://www.bundestag.de/dokumente/protokolle/plenarprotokolle
[5] https://www.bundesgerichtshof.de/DE/Entscheidungen/entscheidungen_node.html
[6] https://pub.cl.uzh.ch/wiki/public/costep/start
[7] https://joint-research-centre.ec.europa.eu/language-technology-resources/dcep-digital-corpus-european-parliament_en
[8] https://www.dnb.de/DE/Professionell/Services/Dissonline/dissonline_node.html
[9] https://researchdata.tuwien.ac.at/records/2zx6e-5pr64
[10] https://pub.cl.uzh.ch/wiki/public/pacoco/medi-notice

Hyper-Parameter	Value(s)
model_type	Unigram | BPE
vocab_size	33k |50k 82k | 100k
character_coverage	0.9999
split_by_number	True
allow_whitespace_only	True
add_dummy_prefix	True
user_symbols	<s>,</s>,<pad>, <eod>, <ph_1>, . . . , <ph_255>
byte_fallback	True
max_sentence_length	4192
normalization_rule_name	NFKC
train_large_corpus	True
remove_extra_whitespaces	False
split_by_whitespace	True

Table 8: Overview of the SentencePiece options that we used for the training of our tokenizers.

medications and pharmaceutical products published by the Swiss Agency for Therapeutic Products.(Graën et al., 2019)

9. Swiss Policy[11] contains documents of the Swiss Legislation Corpus (Höfler and Piotrowski, 2011)

10. OpenSubtitles 2018[12][13] is a collection of translated movie subtitles. (Lison and Tiedemann, 2016)

B Tokenizer

In our experiments, we focused on the *Huggingface tokenizer* library (Moi and Patry, 2023) and the *SentencePiece* library (Kudo and Richardson, 2018). We use the standard settings of the SentencePiece library if not stated otherwise in Table 8. For the HuggingFace tokenizer library Table 9 shows where we deviated from the standard values.

C LLM Architecture and Hyperparameters

Regarding the training architecture of our 2.6B parameter models, we followed closely the architecture of GPT-3 (Brown et al., 2020a). An overview

[11] https://pub.cl.uzh.ch/wiki/public/pacoco/swiss_legislation_corpus

[12] https://opus.nlpl.eu/OpenSubtitles-v2018.php

[13] https://www.opensubtitles.org/de/index.cgi

Hyper-Parameter	Value(s)
model_type	BPE
vocab_size	33k | 50k 82k | 100k
limit_alphabet	512
nfkc_normalizer	True
lowercase_normalizer	False
strip_accents_normalizer	True
pre_tokenizer	ByteLevel, Digits

Table 9: Overview of the Huggingface options that we used for the training of our tokenizers.

Hyper-Parameter	Value
# Hidden Dimension	2560
# Layers	32
# Attention-Heads	32
Sequence-Length	2048
Optimizer	Adam
Adam$-\beta_1$	0.9
Adam$-\beta_2$	0.9
Learning rate	1.6e-4
Learning rate decay	Cosine
Precision	BF16
FlashAttention	2.0
Position-Embeddings	Rotary

Table 10: Overview of the LLM hyperparameters that we used for the training.

of the used architecture details and hyperparameters is given in Table 10.

For training the models, we used a fork of Megatron-LMhttps://github.com/NVIDIA/Megatron-LM.

D Intrinsic Tokenizer Evaluation

Besides studying the overlap of the same algorithm on the same thesaurus, we were also interested in vocabulary overlaps across algorithms and thesauruses see Fig. 5. What we can observe is that multilingual vocabulary and English vocabulary have a rather small overlap between 24% and 34% that remains similar across increasing vocabulary sizes. Across algorithms, we can see that Unigram and BPE of SentencePiece have a slightly higher overlap than Unigram of SentencePiece and BPE of Huggingface. We think this might be due to library-specific preprocessing steps and more similar hyperparameters.

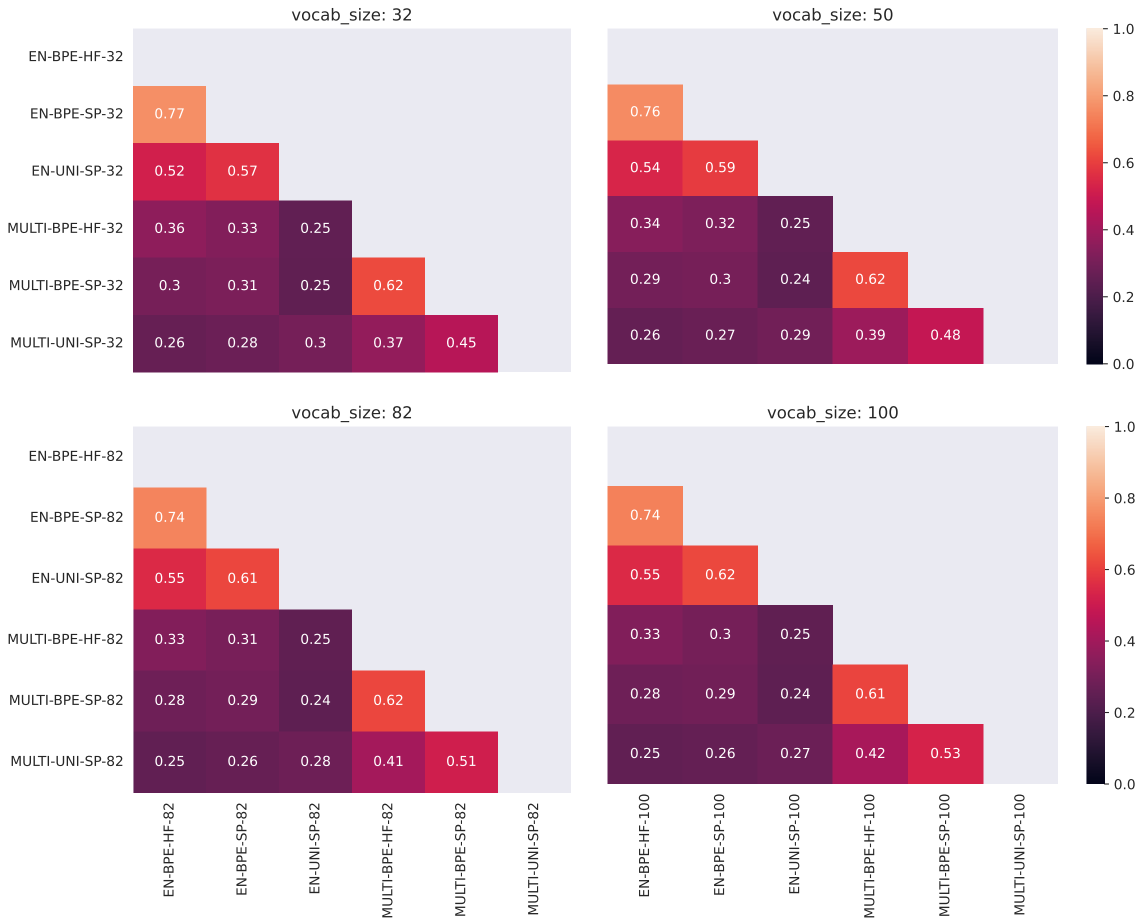


Figure 5: Vocabulary overlap between the examined tokenizers

	Model	Non-English	English	German
	GPT-2-50	3.87	2.58	4.59
EN	BPE-HF-33	3.8	**2.32**	4.52
	BPE-HF-50	3.79	2.38	4.45
	BPE-HF-82	3.88	2.55	4.51
	BPE-HF-100	3.96	2.67	4.58
	BPE-SP-33	3.86	2.37	4.66
	BPE-SP-50	3.89	2.42	4.68
	BPE-SP-82	4.02	2.59	4.78
	BPE-SP-100	4.11	2.71	4.84
	UNI-SP-32	4.01	2.36	4.73
	UNI-SP-50	4.02	2.42	4.75
	UNI-SP-82	4.12	2.59	4.83
	UNI-SP-100	4.21	2.71	4.88
MULTI	BPE-HF-33	2.71	2.46	3.04
	BPE-HF-50	2.7	2.5	3.01
	BPE-HF-82	2.8	2.65	3.09
	BPE-HF-100	2.88	2.76	3.17
	BPE-SP-33	2.68	2.55	2.99
	BPE-SP-50	2.67	2.57	2.95
	BPE-SP-82	2.76	2.72	3.03
	BPE-SP-100	2.85	2.82	3.1
	UNI-SP-33	2.68	2.55	2.94
	UNI-SP-50	**2.66**	2.58	**2.91**
	UNI-SP-82	2.76	2.73	2.99
	UNI-SP-100	2.84	2.83	3.07

Table 11: Computational training costs per word (GFLOPs) for different tokenizers.

D.1 Computational Costs Per Word During Training

Table 11 shows the average computational training costs for processing a word during the forward and backward pass.

E Infrastructure & Computational Costs

We trained each of our 26 2.6B parameter models on NVIDIA A100 GPUs, and the training of each model took up to 2304 GPU hours. Therefore, the total training costs amounted to $\approx$ 59.000 GPU hours.