No Single Tokenizer Feature Reliably Predicts Downstream Language Model Performance

Antonio Lopardo[1,5], Luis Frentzen Salim[1,2,3], Shikhar Srivastava[1,4], Kevin Jiang[1], Mridul Sharma[1], Catherine Arnett[1]
[1]EleutherAI, [2]National Taiwan University of Science and Technology,
[3]Academia Sinica, [4]University of Rochester, [5]University of Basel
Correspondence: catherine@eleuther.ai

Abstract

Tokenization is a critical component of the language modeling pipeline, but it is poorly understood. Tokenizers can impact downstream performance, but there is currently mixed evidence about which tokenizer features predict a better language model without requiring a full training run. In this paper, we analyze approximately 200 language models trained on nine different languages. We evaluate both models and their tokenizers and test whether there is a relationship between any proposed intrinsic tokenizer evaluations and model performance, as measured by a wide variety of benchmarks. Despite the quantity and variety of proposed intrinsic tokenizer metrics, we find that none consistently predicts downstream performance across model families, tasks, and languages. This motivates the need for novel tokenizer evaluation metrics and approaches. We propose novel linguistically grounded metric introduced in this work shows that new metrics can capture aspects of tokenizer quality that existing metrics miss, though their predictive power is likely to vary across model families, languages, and tasks.

1 Introduction

Tokenization is one of the earliest steps in the language modeling pipeline and it affects many aspects of the language model, including performance. Several studies have shown performance differences between models trained otherwise identically, varying only the tokenizer (Ali et al., 2024; Goldman et al., 2024; Schmidt et al., 2024; Lotz et al., 2025). However, it is not currently well understood which properties of a tokenizer generally lead to better performance. This prevents language model developers from making principled design decisions for selecting tokenization algorithm, pretokenizer, vocabulary size, etc.

Reliable *intrinsic tokenizer evaluations*, i.e. metrics derived directly from the tokenizer, would be valuable if they allow researchers to identify more optimal tokenizers before training any language models. Many intrinsic metrics have been proposed, ranging from statistical (e.g. Zipfianness; Lotz et al., 2025), compression (e.g. fertility; Rust et al., 2021), information theoretic (e.g. Rényi entropy; Zouhar et al., 2023), and linguistic (e.g. MorphScore; Arnett and Bergen, 2025; Arnett et al., 2025b) metrics. However, some of these have later been shown to correlate poorly with downstream performance (Schmidt et al., 2024; Cognetta et al., 2024; Arnett and Bergen, 2025; Arnett et al., 2025b).

In the current work, we conduct a comprehensive analysis of proposed intrinsic tokenizer metrics across nine languages and a wide variety of tasks. We use three model suites: two existing suites and a new suite of monolingual models we trained from scratch, which differ only in their tokenizers. We address three related high-level questions:

RQ1: Are there any intrinsic tokenizer metrics that reliably predict downstream model performance? Given the mixed evidence about which tokenizer metrics correlate with better downstream performance, we calculate the correlation between most proposed tokenizer metrics and downstream language model performance for three different model suites. Many of these tokenizer metrics have been proposed based on empirical evidence from a single language or a small number of languages. We ask whether any of these metrics reliably correlate with downstream performance across tasks, languages, and model suites (Section 6). We also propose and test a novel intrinsic tokenizer metric, which we adapt from prior work on entropy-guided segmentation (Section 3).

RQ2: What is the relationship between different tokenizer metrics? Several proposed intrinsic tokenizer metrics are in apparent conflict with one another. For example, MorphScore measures

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 1968–2014
October 24-29, 2026 ©2026 Association for Computational Linguistics

the extent to which a tokenizer preserves meaningful linguistic boundaries in a word. One way tokenizers can achieve higher MorphScore is by segmenting words into more tokens, so that token boundaries are more likely to overlap with linguistic boundaries. This would affect compression, and it is thought segmenting a word into more tokens (i.e. having worse compression) is associated with *worse performance*. We ask: are there tradeoffs between features like MorphScore and compression that an optimal tokenizer should balance? In Section 7, we characterize the relationships between different intrinsic evaluations. We can then ask whether identifying these tradeoffs helps us understand which combinations of metrics might be more useful for predicting downstream performance. We then test whether any combination of intrinsic metrics may more reliably predict downstream performance.

RQ3: Do orthographic tasks correlate with better performance? It has been asserted that there are certain undesirable language model behaviors that are caused by tokenization. One of these is the lack of orthographic awareness, including the so-called "strawberry problem" (Silberling, 2024; Cosma et al., 2025). This refers to the fact that even highly performant language models may not have orthographic awareness of their tokens and, as a result, cannot count the number of *r*s in 'strawberry.' Partially in response to this observed behavior, there have been a number of proposed orthographic knowledge and string manipulation tasks, e.g. LMEntry (Efrat et al., 2023) and CUTE (Edman et al., 2024), which aim to measure how much awareness language models have of the characters in their tokens and their ability to manipulate strings based on that awareness. However, there is minimal theoretical motivation behind these tasks. Do these tasks measure desirable properties of a tokenizer? To address this question, we ask whether performance on orthographic tasks correlates with better performance on other tasks.

Our main findings are:

- **RQ1:** No intrinsic tokenizer metric reliably predicts downstream performance across model suites, languages, and tasks. While idiosyncratic correlations exist, they do not generalize.
- **RQ2:** Intrinsic metrics are highly redundant, metrics within the same category are strongly correlated, and combining metrics does not improve prediction.
- **RQ3:** Orthographic tasks are only weakly correlated with general downstream performance, raising questions about their role in guiding tokenizer design.

Tokenizers are a critical aspect of language model design (Alqahtani et al., 2026), but it is still not known how to measure whether a tokenizer is suitable for a particular model. Our results motivate new approaches to tokenizer design and evaluation.

2 Related Work

Most researchers evaluate tokenizers through intrinsic evaluation: evaluating the tokenizer itself, independent of any model. Reliable intrinsic evaluation would be valuable because experiments involving training models while varying only the tokenizer can be prohibitively expensive. Here we introduce the most widely used intrinsic tokenizer metrics that have been proposed.

Compression metrics. One of the most commonly studied dimensions of tokenizer quality is compression. Compression is measured in a variety of ways, most often fertility (tokens per word; Rust et al., 2021) or bytes per token (Liu et al., 2025). Higher compression on the test set is associated with better performance in machine translation (Gallé, 2019). Delétang et al. (2024) show that minimizing a language model's cross-entropy loss is equivalent to minimizing the number of bits required to compress the data, establishing a direct connection between compression and modeling quality. But in controlled studies where comparable models are trained with different tokenizers, results generally have not supported this link between compression and performance (Ali et al., 2024; Schmidt et al., 2024).

Information-theoretic metrics. Information-theoretic measures have been proposed to characterize the distributional properties of tokenizer vocabularies. Rényi entropy (Zouhar et al., 2023) generalizes Shannon entropy and penalizes unbalanced token distributions, particularly those dominated by a few high-frequency tokens. Other related metrics include unigram and bigram entropy, which measure the diversity of token usage and contextual variety where, similarly, higher entropy may be more desirable. However,

Cognetta et al. (2024) showed that while Rényi entropy captures some desirable properties of a tokenizer, it is not a holistic measure of tokenizer quality. Furthermore, Dagan et al. (2024) find that higher Rényi entropy in fact is correlated with *lower* performance in code generation. Beyond vocabulary characterization, entropy at character boundaries can identify meaningful linguistic units, a principle used by the Byte Latent Transformer to dynamically segment byte sequences (Pagnoni et al., 2025), and may help preserve multi-word expressions, an important aspect of tokenizer quality (Gow-Smith et al., 2022, *inter alia*).

Linguistic metrics. Tokenizers that preserve the internal structure of words and segment along morphological boundaries are argued to lead to better performance (Klein and Tsarfaty, 2020; Park et al., 2020; Vasiu and Potolea, 2020; Hofmann et al., 2020; Nzeyimana and Niyongabo Rubungo, 2022; Libovický and Helcl, 2024; Bauwens and Delobelle, 2024). For example, Bostrom and Durrett (2020) proposed evaluating tokenizers based on their alignment with morphological structure. Subsequent work has further developed morphology-based evaluation criteria, showing, for instance, that derivational segmentation improves model interpretations of complex words (Hofmann et al., 2021). But, there are also a number of studies which have failed to find benefits of morphologically aligned tokenization (Macháček et al., 2018; Saleva and Lignos, 2021; Choo and Kim, 2023; Toraman et al., 2023; Arnett et al., 2024b; Arnett and Bergen, 2025; Arnett et al., 2025b).

Statistical metrics. Statistical approaches evaluate how effectively tokenizers use their vocabularies and how token usage is distributed. Measures like vocabulary utilization (the proportion of vocabulary items actually used when encoding a corpus) and type-token ratio (the ratio of unique tokens to total tokens) capture basic aspects of vocabulary coverage and lexical diversity. Other metrics capture how token usage is distributed across the vocabulary: average (token) rank is the expected rank of a token sampled from a corpus when vocabulary items are sorted by frequency. Limisiewicz et al. (2023) show that vocabulary allocation, which they measure via average token rank and characters per token, has nuanced effects on multilingual performance: higher allocation benefits word-level tasks but harms masked language modeling scores. Also examining vocabulary frequency distributions, Lotz et al. (2025) propose metrics based on Zipf's law, which assess whether token frequencies follow the power-law patterns observed in natural language. In these studies, these metrics correlate more strongly with downstream performance than compression-based metrics, particularly in multilingual settings.

3 U-Statistic

Here we introduce a novel intrinsic tokenizer metric. We adopt the U-statistic of Hu et al. (2025), a per-token score that formalizes ideas from unsupervised word segmentation (Harris, 1955; Jin and Tanaka-Ishii, 2006), and aggregate it over a tokenizer's vocabulary as an intrinsic evaluation asking: *does this token correspond to a natural linguistic unit?* The score, for each token w, combines two signals from corpus character statistics:

$$U_{\text{stat}}(w) = \underbrace{\text{PMI}_{\min}(w)}_{\text{internal cohesion}} + \lambda \cdot \underbrace{\min\big(H_{\text{left}}(w),\, H_{\text{right}}(w)\big)}_{\text{contextual separability}} \quad (1)$$

The first term takes the minimum pointwise mutual information (PMI) over adjacent character pairs inside the token, so a single weak pair flags the token as crossing a natural boundary. The second takes the minimum branching entropy at the token's edges; high entropy signals a natural break point (Harris, 1955). We report mean (`u_mean`), median (`u_median`), frequency-weighted mean (`u_mean_weighted`), and log-frequency-weighted mean (`u_mean_log_weighted`) of U_{stat}; full definitions and illustrative examples are in Appendix A.

4 Materials

4.1 Models

We use three suites of models: the models from TokSuite (Altıntaş et al., 2025); the models in the paper "Tokenization Is More Than Compression" (Schmidt et al., 2024; henceforth TIMTC); and a suite of models we train ourselves, VISTA.

TokSuite models. The TokSuite models (Altıntaş et al., 2025) consist of 14 1B-parameter Llama architecture models with identical training budget and initialization, which differ only in their tokenizers. These models enable us to attribute performance differences to tokenizer characteristics. The tokenizers are drawn from existing models

and feature various vocabulary construction algorithms (BPE, Unigram, WordPiece), as well as both byte-level and subword tokenizers. Among these models, vocabulary sizes vary substantially, from a byte-level vocabulary of 259 tokens to approximately 250k (most are between 30k–130k).

The models were trained on a multilingual corpus consisting of 40B tokens of English data from FineWeb-Edu (Penedo et al., 2024) and 15B tokens each of Turkish, Chinese, Italian, and Farsi data from FineWeb-2 (Penedo et al., 2025).

TIMTC models. In Schmidt et al. (2024), the authors investigate the hypothesis that the effectiveness of BPE arises primarily from its text compression capabilities. They release a suite of 54[1] 350M-parameter models that differ only in their tokenizers, and which vary systematically in their: (1) *vocabulary construction*, how the vocabulary is built; (2) *initial vocabulary*, the token set used to seed vocabulary construction; (3) *pre-tokenization*, rules that restrict or enforce token boundaries; and (4) *segmentation*, the algorithm that assigns segmented text into tokens given a vocabulary (see Appendix B.1 for details). The models are trained only on English data.

This carefully controlled suite—where models differ only in their tokenization choices while sharing identical architecture, training data, and hyperparameters—allows us to isolate the effects of tokenization on downstream performance.

VISTA Models. We introduce VISTA (Vocabulary-size Investigation Suite for Tokenizer Analysis), a suite of monolingual models for five languages (English, Thai, Urdu, Amharic, and Vietnamese), which systematically varies vocabulary size using both BPE and Unigram tokenizers. For each language and vocabulary size, we pre-train a language model based on the LLAMA 1 architecture (Touvron et al., 2023). The language models are pre-trained using a standard auto-regressive cross-entropy loss. We train monolingual models with corresponding language-specific tokenizers, with varying vocabulary sizes. Varying vocabulary size is one way to systematically manipulate tokenizer characteristics like compression (Arnett et al., 2025a) and has been shown to impact model performance (Toraman et al., 2023).

[1]We used 45 of the available models due to persistent out-of-vocabulary errors that we could not resolve.

We train two model sizes: 130M and 350M parameters. We detail the hyperparameters and training design decisions in Appendix B.3. We release these models on Hugging Face[2][3] along with their training pipeline[4].

Training Data For the 130M parameter models, we used data from `fish-food`[5] (Chang et al., 2026), with each language's training corpus sized to approximately 1 GB $\times$ its byte premium (or 'BP', as discussed below). Each tokenizer is trained on the equivalent of 300 MB of English data in the corresponding language (Arnett et al., 2025a). For the larger 350M models, we use FineWeb and FineWeb-2 (Penedo et al., 2024, 2025), with a Chinchilla optimal (Hoffmann et al., 2022) 7 billion tokens ($\sim$31 GiB tokenized) for English. We control for character-encoding efficiency, following Chang et al. (2026), by normalizing the corpus size according to the *byte premium* (BP), i.e. the ratio of UTF-8 bytes required to encode content-matched text in a language L vs. English (Arnett et al., 2024a). Each non-English language's training data is further byte premium (BP)-scaled to match: target_bytes $=$ BP $\times$ (English ref. bytes), yielding 42–85 GiB of tokenized data depending on the language and vocabulary size (see Table 4). The 130M models are trained for 10 epochs over their BP-scaled sets, matching the multi-epoch regime advocated for smaller monolingual budgets (Muennighoff et al., 2023; Chang et al., 2026). The 350M models are trained for 1 epoch on their BP-scaled Chinchilla-optimal sets, except Urdu which uses 3–4 epochs to compensate for limited FineWeb-2 availability ($\sim$24% of the BP-scaled target). We do not train an Amharic 350M model, because FineWeb-2 contains less than 5% of the data necessary to optimally train a model of that scale. VISTA architecture and training details are discussed in detail in Appendix B.3.

Tokenizer Manipulations For each language, we train tokenizers from two families: (i) BPE and (ii) Unigram. For each language and tokenizer family, we train tokenizers with vocabulary sizes $v \in \{8192, 16384, 32768, 49152, 65536, 81920,$

[2]https://huggingface.co/collections/shikhar-srivastava/vista-350m

[3]https://huggingface.co/collections/shikhar-srivastava/vista-130m

[4]https://github.com/shikhar-srivastava/vista-training/

[5]https://huggingface.co/datasets/goldfish-models/fish-food

98304, 114688}, following Arnett et al. (2025a). The language modeling head dimensions are adjusted to v, with all other architectural and optimization hyperparameters held fixed (see Appendix Table 7). This design isolates the effect of language and vocabulary size on compression while controlling for compute and model capacity. For each language, we train a model for each combination of tokenizer type and vocabulary size. Due to computational resource constraints, at the larger scale, we train only on two of the vocabulary sizes: 8192 and 98304.

4.2 Intrinsic Tokenizer Metrics

We use 27 previously proposed intrinsic tokenizer metrics from the different categories discussed in Section 2 and the two U-statistic metrics we introduced in Section 3. We list the metrics used in Appendix A and provide additional description of each. To compute these metrics, we use the `tokenizer-intrinsic-evals` (TokEval)[6] suite (Meister, 2026), a unified framework for tokenizer evaluation that provides implementations of information-theoretic, morphological, and statistical metrics, among others. For metrics not in the evaluation suite, we implemented them in our own fork (marked with † in Table 2, Appendix A). The fork is available online, linked from our main repository.[7] For test data we use samples from the OSCAR corpus (Abadji et al., 2022).

4.3 Model Evaluations

To quantify the performance impact of different tokenization processes, we carefully selected a group of downstream task benchmarks. In total, we use 48 distinct benchmarks spanning linguistic acceptability, natural language understanding, commonsense reasoning, world knowledge and question answering, multi-task suites, token-level (orthographic) tasks, and perplexity, across nine languages; the per-suite lists are given in Tables 8, 9, and 10. We use the LM Evaluation Harness (Gao et al., 2024; Biderman et al., 2024) for evaluation. For the tasks not implemented in the harness, we implemented them ourselves. This also enabled us to adapt the harness to support custom models, especially the TIMTC models which use non-standardized tokenizers. We selected the benchmarks using a few criteria: first, we chose tasks that were appropriate for the scale of models we evaluate in these experiments (100M-1B parameters). Second, we selected tasks that we expect to show sensitivity to tokenization variation, particularly linguistic benchmarks like BLiMP and other NLU tasks. Finally, we select an additional set of common tasks such as commonsense reasoning and question answering. The full list of benchmark tasks is provided in Appendix C.

5 Tokenizers *Do* Affect Model Performance

Before addressing our research questions, we describe the results from the tokenizer and model evaluation. We release the raw tokenizer evaluation metrics and language model benchmark scores on GitHub7[0] and the full model evaluation results are presented in Appendix D.

TokSuite models. Consistent with previous work, we find that across multiple model suites, many evaluations, and several languages, the tokenizer used to train a model significantly impacts performance. The range of aggregate performance was 4% with individual task performance differing by up to 50% (for ANALOGICAL, Wijesiriwardene et al., 2023; the lowest performance was 37% and the highest was 89%).

Some of these differences may be driven by the vocabulary size, which also impacts the number of embedding parameters, and therefore model size. We find a moderate positive correlation between vocabulary size and average performance for these models ($\rho = 0.64, p < 0.01$). But vocab size doesn't account for all differences. XGLM, BLOOM, and Gemma all have vocab sizes of around 256,000. The XGLM tokenizer yields some of the best performance, while Gemma leads to more average performance in aggregate. While these models differ in many aspects, this suggests that other tokenize features besides vocabulary size may drive model performance more.

TIMTC models. After evaluating TIMTC on our own set of evaluations, we find that there are differences in performance between the models in the suite, as found in Schmidt et al. (2024). Across the different tokenizer conditions, average performance differed by up to 6.5% across models. The maximum range in performance per task was 37%.

[6] https://github.com/cimeister/tokenizer-intrinsic-evals

[7] https://github.com/catherinearnett/tokenizer-metric-multilingual-correlations

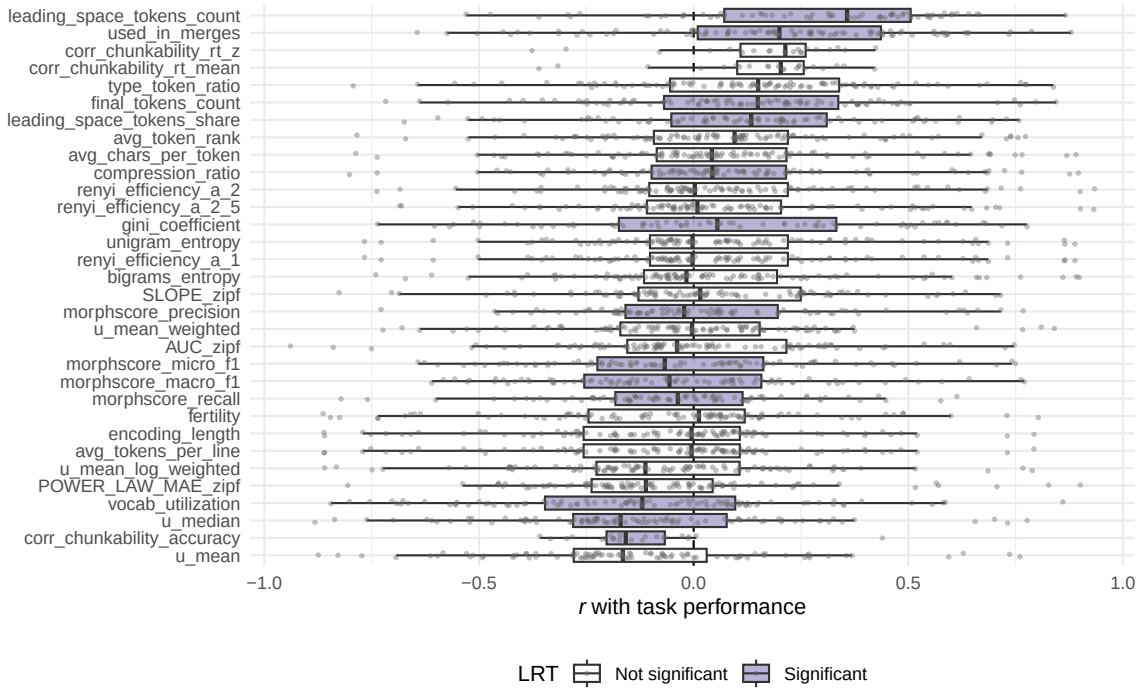


Figure 1: Each row represents the distribution of correlation coefficients (x-axis) between a tokenizer metric (y-axis) and individual tasks. The color of the box plot represents whether the overall correlation between the tokenizer metric and aggregate performance is significant.

VISTA models. The VISTA models also exhibit differences in performance depending on tokenizer, with differences in aggregate performance of 4%-26%, depending on the language. Unlike the TokSuite models, however, we do not find any significant relationship between vocabulary size and model performance ($p = 0.407$; results in Appendix D.3.1). One reason for this difference could be that while TokSuite tokenizers differ in many aspects, VISTA tokenizers only differ in vocabulary size.

6 No Intrinsic Tokenizer Metric Reliably Predicts Downstream Performance

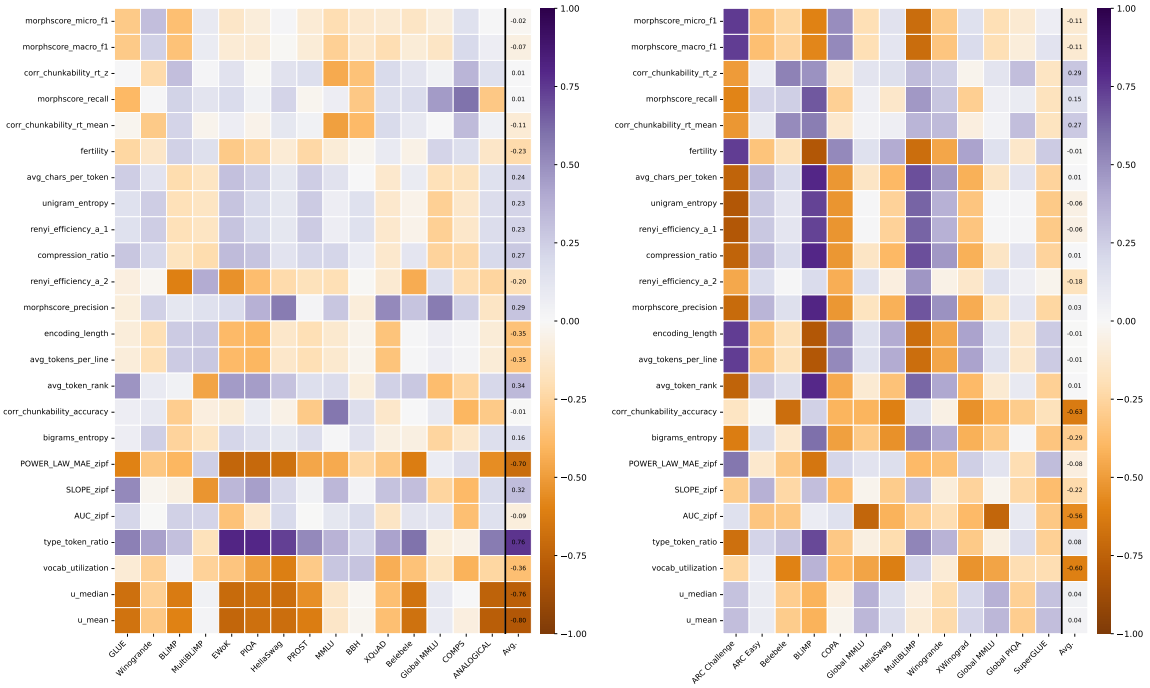
Analysis method. To answer this question, we test for a correlation between model performance and individual intrinsic tokenizer metrics, following Ali et al. (2024). In the main analysis, we test correlation with task performance, which we normalized by first z-scoring the accuracy scores within each language-task combination (e.g. the Turkish subset of Global MMLU). We test for a correlation across all task scores. We report all correlations between individual metrics and tasks in Appendix E.

We find that for several of the tasks that we evaluate our models on, a large number of models perform at- or near-chance, especially the smaller models. We conduct an additional analysis after excluding these tasks, which we describe in Appendix F. The results are qualitatively the same.

Main results. We find that 13 of the 29 tokenizer metrics are significantly correlated with z-scored model performance, across all tasks, languages, and model suites. The metrics with the strongest correlations are `leading_space_tokens_count` ($r = 0.27$, $p < 0.001$)[8], `used_in_merges` ($r = 0.24$, $p < 0.001$), and `leading_space_tokens_share` ($r = 0.22$, $p < 0.001$). These are relatively weak correlations.

We found that each metric had a wide range of correlation strengths and directions across all tasks and languages. While there are instances where there are strong correlations between a metric and an individual task, no patterns hold across all languages, tasks, and model suites. In Figure 1, we plot the distribution of correlations between each tokenizer metric and individual tasks. We note that almost all tokenizer metrics (including the three metrics with the strongest overall correlations) have at least one task for which there is a

[8] All p-values in this paper have a Bonferroni correction for multiple comparisons, unless otherwise noted. We report corrected and uncorrected p-values in Appendix E.



(a) TokSuite

(b) VISTA

Figure 2: Correlations between intrinsic metric and English model performance for two model suites, including aggregated performance (Avg.). The color of each cell corresponds to direction of correlation and the shade corresponds to the strength. Results for other languages are reported in Appendix E.

moderate ($r > 0.5$) positive correlation and another task for which there is a moderate negative correlation. Not only do the magnitude of the correlations vary by task, but also the direction.

We test how much of the variance can be accounted for simply by model size (in number of total parameters), and find that while for 23 of the metrics, it significantly improves model fit for a linear regression above and beyond model size, the maximum increased variance explained is negligible ($R^2 = 0.021$, $p < 0.001$). The full results of this analysis can be found in Table 27.

If we look at the correlations between metrics and tasks within a single language and a single model suite, there are some trends. Figure 2 shows the correlations between each tokenizer metric and both individual and aggregated English tasks for the TokSuite and English VISTA models. For example, U-statistic metrics are strongly positively correlated with downstream performance in English for the TokSuite models across most tasks.

Comparing within model suite but across languages (cf. Figure 2, left, and Figure 3), the strength of the correlations varies greatly but the direction of the correlation largely holds in aggregate. Considering individual tasks, the correlations

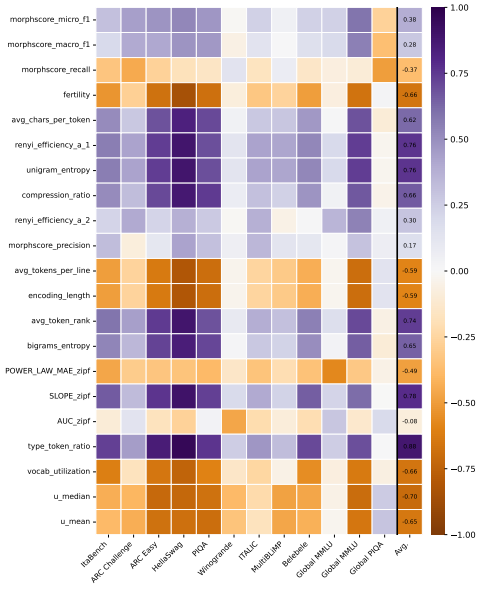


Figure 3: Correlation between each tokenizer metric and the Italian downstream evaluations, individually and in aggregate (Avg.) for the TokSuite models.

are not consitent. For example `SLOPE_zipf` is positively correlated with MultiBLiMP in English, but negatively correlated in Italian for the TokSuite models. Comparing across model suites, the cor-

relations can change again, e.g. `SLOPE_zipf` is negatively correlated with MultiBLiMP in English with the VISTA models (Figure 2, right). This supports the aggregate analysis that the correlations are unstable across models, tasks, and languages.

We report the by-language, by-task correlations for each model suite in Figures 8, 9, and 10 in the Appendix.

6.1 U-statistic

Among TokSuite models, `u_mean` and `u_median` are the top-correlated metrics with aggregate English and Turkish performance (Figure 2 and Figure 8c), outperforming all previously proposed intrinsic metrics. For TIMTC, boundary statistics remain among the more strongly correlated metrics, though with weaker effect sizes. However, for VISTA, U-statistics show almost no relationship with performance. All three suites control for architecture, yet the U-statistic's predictive power varies substantially across them, suggesting that the relationship between linguistic alignment and downstream performance depends on other properties of training.

The metric is also language-dependent (Table 1). Alphabetic languages cluster around $\bar{u} \approx 2.4$, while Chinese and Farsi are $\sim 2.7\times$ higher ($\bar{u} \approx 6.4$) and show no correlation with performance. Decomposing the U-statistic into its components explains this gap. The PMI_{min} component varies more across languages than across tokenizers: cross-tokenizer standard deviations are small across all languages (0.09–0.34), but the mean values diverge sharply, alphabetic languages have negative PMI_{min} (-1.06 to -1.70), while Farsi and Chinese have positive PMI_{min} (3.09 and 1.93). In alphabetic scripts, most adjacent character pairs within tokens have moderate cohesion, so the PMI_{min} bottleneck is sensitive to tokenization quality: a bad split produces a low-PMI pair that drags the score down. In Chinese, characters are meaning-bearing units that combine productively, yielding uniformly high character-pair PMI; the bottleneck rarely binds, so all tokenizers score similarly. Boundary entropy (BE) is more variable across tokenizers (std 0.26–0.50). Hu et al. (2025) originally developed these signals for Chinese pre-tokenization, where local contrasts in PMI and entropy effectively identify word boundaries, but aggregated across a tokenizer's full vocabulary, these contrasts wash out. Token frequencies-weighted variants (`u_mean_weighted`, `u_mean_log_weighted`) do not resolve this, as the issue lies in the underlying character statistics rather than the aggregation; language-specific tuning of λ could help if the two components degrade unevenly across scripts, but we leave this to future work.

Table 1: U-statistic components by language for the TokSuite models (mean and std across the 13 TokSuite tokenizers, excluding ByT5, whose byte-level vocabulary has no character statistics). PMI_{min} shows minimal cross-tokenizer variation, especially for Chinese (std 0.09), confirming it reflects script properties rather than tokenizer quality.

Language	**PMI_{min} (std)**	**BE (std)**	$\bar{u}$
English	-1.06 (0.22)	0.77 (0.30)	2.44
Italian	-1.42 (0.20)	0.81 (0.26)	2.42
Turkish	-1.70 (0.21)	0.84 (0.30)	2.35
Farsi	3.09 (0.34)	0.76 (0.41)	6.31
Chinese	1.93 (0.09)	1.11 (0.50)	6.51

The U-statistic captures a complementary dimension of morphological alignment, whether tokens correspond to coherent character sequences with natural boundaries, using only unsupervised character statistics, without requiring morphological annotations. In settings where it has discriminative power, it outperforms previously proposed metrics. However, its predictive power varies across model suites and breaks down for writing systems where character-level co-occurrence statistics lack the structure the metric relies on. These results suggest that linguistically grounded metrics are a promising direction for tokenizer evaluation, but that their effectiveness is likely to depend on model family and language.

6.2 Summary

Together these results show the lack of consistency of the tokenizer metrics to reliably predict model performance across different model suites, tasks, and languages. Because the results are not only different between languages but also between models, we argue that none of the intrinsic tokenizer evaluations is able to reliably predict model performance.

7 Combining Metrics Does Not Improve Prediction

Here we ask whether combining multiple metrics can better predict downstream performance. We find that many of the metrics are correlated with each other, which suggests that many of the metrics

may be identifying similar or related tokenizer features. We test whether any combination of metrics better predicts performance across all languages. We fit linear regressions with all combinations of two metrics. We find that although combinations significantly improve model fit, the joint R^2 never exceeds 0.09, and the gain in R^2 never exceeds 0.01. Therefore, combining two intrinsic tokenizer evaluations does not seem to meaningfully help predict model performance. Additional results and statistical tests are reported in Appendix E.2.

8 Orthographic Tasks Are Weak Predictors

Here we analyze whether there is a relationship between orthographic tasks and general model performance. In this analysis, we focus on the TokSuite models, as the other models achieve near-zero performance at most of these tasks. We test whether the orthographic tasks are correlated with any of the downstream tasks or aggregated model performance. Only TokSuite (English) shows a significant correlation with downstream task performance ($r = 0.341, p = 0.029$), though the correlation is relatively weak. The other tasks are all not significantly correlated. We report the results in Appendix E.1. These results call into question whether these tasks should be used to shape tokenizer selection and design.

9 Discussion

Lack of Robust Correlations. While some tokenizer metrics correlate with individual tasks for some models, there is no clear pattern. We demonstrated that tokenizers affect model performance (Section 5), but none of the proposed tokenizer evaluations correlates with performance. We find that using two metrics instead of one does not improve predictive power. Based on these results, we would predict that testing for a correlation between a tokenizer metric and performance on a small number of tasks—especially in a single language or in one family of models—may often result in a significant result. However, our analysis suggests that those results are unlikely to generalize to additional tasks, languages, and model families.

The current approach to intrinsic tokenizer evaluation currently does not allow us to determine which tokenizer will result in a better model. More work is needed to be able to determine what an optimal tokenizer looks like and how to identify it.

Different Approaches To Tokenizer Evaluation. We have focused on the impact of tokenization on model performance. There are other ways in which tokenizers affect language models. Tokenizers in part determine the speed and cost of model training (Ahia et al., 2023; Petrov et al., 2023; Dagan et al., 2024; Ali et al., 2024; Asgari et al., 2025), because they determine how many computations are needed for a model to process a given text. Therefore compression may be a useful metric, even if it does not correlate with downstream performance. Relatedly, tokenizer evaluation could help identify unused or under-used tokens in the vocabulary that may lead to undertrained tokens and subsequently undesirable model behavior (Land and Bartolo, 2024).

Another dimension is multilingual parity. Ideally, tokenizers can contribute to more balanced multilingual performance (Petrov et al., 2023; Ahia et al., 2023). By better understanding how tokenization impacts languages differently (e.g. Arnett et al., 2025a), we can find ways to develop more balanced tokenizers (e.g. Limisiewicz et al., 2024; Foroutan et al., 2026).

The Impact of Tokenizer Design Decisions. We do not exhaustively analyze the effect of different tokenizer design decisions, but we do discuss the effect of vocabulary size in Section 5. Vocabulary size is one of the parameters that can be set for a tokenizer, but which is usually chosen with little rationale (Alqahtani et al., 2026). Tao et al. (2024) find that most model scales benefit from increased vocabulary, suggesting a bigger-is-better principle for vocabulary size. However, for the small models used in this study, at a scale where changes in vocabulary size have a large proportional impact on the number of trainable parameters, we do not find a robust effect across our model suites. Alongside developing better tokenizer evaluations, we need to develop a better understanding of how different design decisions impact model performance.

10 Conclusion

We have conducted, to the best of our knowledge, the most comprehensive study about the relationship between intrinsic tokenizer metrics and downstream model performance. We find that none of these intrinsic measures reliably predicts model performance and we call for a novel approach to tokenizer evaluation.

Limitations

Each of the model suites we use has its own limitations. TokSuite was trained on tokenizers from existing models, which vary in a number of dimensions: each was trained on a different mix of data, the tokenizers vary in vocabulary size, and many aspects of the tokenization algorithm are different including pretokenization and normalization. Therefore in our analyses, we are unable to completely account for each of these variables which may influence the resulting model's performance. The TIMTC models include only models trained on English, therefore we are unable to determine to what extent any patterns may generalize to other languages. Finally, the VISTA models are trained in a controlled way and are trained for multiple languages. However, they are much smaller than the models in the other suites. None of these model suites is able to definitively address the research questions because of these limitations. However, putting these models together and comparing across suites helps us to address some of the weaknesses of the models.

In our study, we studied models of a variety of sizes, including models up to 1B parameters. That said, our smaller models are generally not very strong. They achieved near-chance performance at several tasks. Analyzing models with poor performance may not provide the same signal as larger models trained on more tokens may provide. We address the effects of poor task performance on our results in Appendix F. Additionally, we highlight that small differences in performance at smaller scales still represent differences in performance that intrinsic tokenizer evaluation should seek to predict and explain.

The benchmarks we used to evaluate model performance varied across languages. Some languages had more available benchmarks than others, and these benchmarks vary in size and quality. Additionally, some of the intrinsic tokenizer evaluations relied on annotated data. The chunkability data from Beinborn and Pinter (2023) is only available in English. There is also not sufficient labeled data to calculate MorphScore for Vietnamese. The lack of evaluations in languages other than English is a widespread issue in the field and limits our analyses.

Acknowledgements

We thank the Center for Integrated Research Computing (CIRC) at the University of Rochester and CoreWeave for providing the computational resources used in this work. We would like to thank Craig Schmidt for making the TIMTC model checkpoints freely available and for his assistance in setting up the PathPiece library. .

Ethical Considerations

This work aims to improve the understanding of the impact of tokenization across multiple settings, including multiple languages. We are not aware of any potential risks or negative impacts of this work. In fact, we hope this work improves the understanding of how design decisions may not affect all languages equally.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4344–4355, Marseille, France. European Language Resources Association.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do All Languages Cost the Same? Tokenization in the Era of Commercial Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923, Singapore. Association for Computational Linguistics.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, Charvi Jain, Alexander Weber, Lena Jurkschat, Hammam Abdelwahab, Chelsea John, Pedro Ortiz Suarez, Malte Ostendorff, Samuel Weinbach, Rafet Sifa, and 2 others. 2024. Tokenizer Choice For LLM Training: Negligible or Crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924, Mexico City, Mexico. Association for Computational Linguistics.

Sawsan Alqahtani, Mir Tafseer Nayeem, Md Tahmid Rahman Laskar, Tasnim Mohiuddin, and M Saiful Bari. 2026. Stop Taking Tokenizers for Granted: They Are Core Design Decisions in Large Language Models. *arXiv preprint arXiv:2601.13260*.

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

arXiv preprint arXiv:2512.20757. Under review at ICLR 2026.

Hossein Amirkhani, Mohammad AzariJafari, Zohreh Pourjafari, Soroush Faridan-Jahromi, Zeinab Kouhkan, and Azadeh Amirak. 2020. FarsTail: A Persian Natural Language Inference Dataset. *arXiv preprint arXiv:2009.08820*.

Catherine Arnett and Benjamin Bergen. 2025. Why do language models perform worse for morphologically complex languages? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6607–6623, Abu Dhabi, UAE. Association for Computational Linguistics.

Catherine Arnett, Tyler A. Chang, and Benjamin Bergen. 2024a. A bit of a problem: Measurement disparities in dataset sizes across languages. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 1–9, Torino, Italia. ELRA and ICCL.

Catherine Arnett, Tyler A. Chang, Stella Biderman, and Benjamin K. Bergen. 2025a. Explaining and mitigating crosslingual tokenizer inequities. In *Proceedings of the Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Catherine Arnett, Marisa Hudspeth, and Brendan O'Connor. 2025b. Evaluating morphological alignment of tokenizers in 70 languages. In *Proceedings of the ICML 2025 Workshop on Tokenization (TokShop)*.

Catherine Arnett, Pamela D. Rivière, Tyler A. Chang, and Sean Trott. 2024b. Different tokenization schemes lead to comparable performance in Spanish number agreement. In *Proceedings of the 21st SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 32–38, Mexico City, Mexico. Association for Computational Linguistics.

Stéphane Aroca-Ouellette, Cory Paik, Alessandro Roncone, and Katharina Kann. 2021. PROST: Physical reasoning about objects through space and time. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4597–4608, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. 2025. MorphBPE: A Morpho-Aware Tokenizer Bridging Linguistic Complexity for Efficient LLM Training Across Morphologies. *arXiv preprint arXiv:2502.00894*.

Sajjad Ayoubi and Mohammad Yasin Davoodeh. 2021. PersianQA: a dataset for Persian Question Answering. `https://github.com/SajjjadAyobi/PersianQA`.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand. Association for Computational Linguistics.

Ezgi Başar, Francesca Padovani, Jaap Jumelet, and Arianna Bisazza. 2025. TurBLiMP: A Turkish benchmark of linguistic minimal pairs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 16495–16510, Suzhou, China. Association for Computational Linguistics.

Thomas Bauwens and Pieter Delobelle. 2024. BPE-knockout: Pruning Pre-existing BPE Tokenisers with Backwards-compatible Morphological Semi-supervision. 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 5810–5832, Mexico City, Mexico. Association for Computational Linguistics.

Lisa Beinborn and Yuval Pinter. 2023. Analyzing cognitive plausibility of subword tokenization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4478–4486, Singapore. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, and 1 others. 2024. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7432–7439. AAAI Press.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

Emrah Budur, Rıza Özçelik, Tunga Gungor, and Christopher Potts. 2020. Data and Representation for Turkish Natural Language Inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8253–8267, Online. Association for Computational Linguistics.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2026. Goldfish: Monolingual

language models for 350 languages. In *Proceedings of the 15th Language Resources and Evaluation Conference (LREC)*.

Tyler A. Chang, Catherine Arnett, and 1 others. 2025. Global PIQA: Evaluating Physical Commonsense Reasoning Across 100+ Languages and Cultures. *Preprint*, arXiv:2510.24081.

Sanghyun Choo and Wonjoon Kim. 2023. A study on the evaluation of tokenizer performance in natural language processing. *Applied Artificial Intelligence*, 37(1):2175112.

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:1803.05457v1*.

Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. 2009. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703.

Marco Cognetta, Vilém Zouhar, Sangwhan Moon, and Naoaki Okazaki. 2024. Two Counterexamples to Tokenization and the Noiseless Channel. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16897–16906, Torino, Italia. ELRA and ICCL.

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.

Adrian Cosma, Stefan Ruseti, Emilian Radoi, and Mihai Dascalu. 2025. The Strawberry Problem: Emergence of Character-level Understanding in Tokenized Language Models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28252–28263, Suzhou, China. Association for Computational Linguistics.

Gautier Dagan, Gabriel Synnaeve, and Baptiste Rozière. 2024. Getting the Most out of Your Tokenizer for Pre-Training and Domain Adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 9784–9805. PMLR.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. 2024. Language Modeling Is Compression. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Lukas Edman, Helmut Schmid, and Alexander Fraser. 2024. CUTE: Measuring LLMs' Understanding of Their Tokens. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3017–3026, Miami, Florida, USA. Association for Computational Linguistics.

Lukas Edman, Helmut Schmid, and Alexander Fraser. 2025. EXECUTE: A Multilingual Benchmark for LLM Token Understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1878–1887, Vienna, Austria. Association for Computational Linguistics.

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

Farhan Farsi, Sadra Sabouri, Kian Kashfipour, Soroush Gooran, Hossein Sameti, and Ehsaneddin Asgari. 2024. SynTran-fa: Generating Comprehensive Answers for Farsi QA Pairs via Syntactic Transformation. Preprint.

Negar Foroutan, Clara Meister, Debjit Paul, Joel Niklaus, Sina Ahmadi, Antoine Bosselut, and Rico Sennrich. 2026. Parity-aware byte-pair encoding: Improving cross-lingual fairness in tokenization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7514–7538. Association for Computational Linguistics.

Matthias Gallé. 2019. Investigating the Effectiveness of BPE: The Power of Shorter Sequences. 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 1375–1381, Hong Kong, China. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The Language Model Evaluation Harness.

Omer Goldman, Avi Caciularu, Matan Eyal, Kris Cao, Idan Szpektor, and Reut Tsarfaty. 2024. Unpacking Tokenization: Evaluating Text Compression and its Correlation with Model Performance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2274–2286, Bangkok, Thailand. Association for Computational Linguistics.

Edward Gow-Smith, Harish Tayyar Madabushi, Carolina Scarton, and Aline Villavicencio. 2022. Improving Tokenisation by Alternative Treatment of Spaces. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*,

pages 11430–11443, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zellig S Harris. 1955. From phoneme to morpheme. *Language*, 31(2):190–222.

Linyang He, Ercong Nie, Sukru Samet Dindar, Arsalan Firoozi, Adrian Florea, Van Nguyen, Corentin Puffay, Riki Shimizu, Haotian Ye, Jonathan Brennan, Helmut Schmid, Hinrich Schütze, and Nima Mesgarani. 2025. XCOMPS: A Multilingual Benchmark of Conceptual Minimal Pairs. In *Proceedings of the 7th Workshop on Research in Computational Linguistic Typology and Multilingual NLP (SIGTYP 2025)*, pages 75–81, Vienna, Austria. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *The Ninth International Conference on Learning Representations (ICLR)*. ArXiv:2009.03300.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2020. DagoBERT: Generating Derivational Morphology with a Pretrained Language Model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3848–3861, Online. Association for Computational Linguistics.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre Is Not Superb: Derivational Morphology Improves BERT's Interpretation of Complex Words. 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 3594–3608, Online. Association for Computational Linguistics.

Yifan Hu, Ningyue Liang, Dachuan Zhao, Jonathan Geuter, Varshini Reddy, Craig W Schmidt, and Chris Tanner. 2025. Entropy-Driven Pre-Tokenization for Byte-Pair Encoding. In *Tokenization Workshop*.

Dieuwke Hupkes and Nikolay Bogoychev. 2025. MultiLoKo: A Multilingual Local Knowledge Benchmark for LLMs Spanning 31 Languages. *Preprint*, arXiv:2504.10356.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R.T. Pramod, Gabriel Grand, Vivian Paulun, Maria Ryskina, Ekin Akyürek, Ethan Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2025. Elements of World Knowledge (EWoK): A Cognition-Inspired Framework for Evaluating Basic World Knowledge in Language Models. *Transactions of the Association for Computational Linguistics*, 13:1245–1270.

Zhihui Jin and Kumiko Tanaka-Ishii. 2006. Unsupervised Segmentation of Chinese Text by Use of Branching Entropy. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 428–435, Sydney, Australia. Association for Computational Linguistics.

Jaap Jumelet, Leonie Weissweiler, Joakim Nivre, and Arianna Bisazza. 2026. MultiBLiMP 1.0: A massively multilingual benchmark of linguistic minimal pairs. *Transactions of the Association for Computational Linguistics*, 14:193–216.

Emmanuel Keuleers, Paula Lacey, Kathleen Rastle, and Marc Brysbaert. 2012. The British Lexicon Project: Lexical decision data for 28,730 monosyllabic and disyllabic English words. *Behavior Research Methods*, 44(1):287–304.

Stav Klein and Reut Tsarfaty. 2020. Getting the ##life out of living: How Adequate Are Word-Pieces for Modelling Complex Morphology? In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 204–209, Online. Association for Computational Linguistics.

Sander Land and Max Bartolo. 2024. Fishing for Magikarp: Automatically Detecting Under-trained Tokens in Large Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11631–11646, Miami, Florida, USA. Association for Computational Linguistics.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2024. CMMLU: Measuring Massive Multitask Language Understanding in Chinese. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11260–11285, Bangkok, Thailand. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 2024. Lexically Grounded Subword Segmentation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7420, Miami, Florida, USA. Association for Computational Linguistics.

Tomasz Limisiewicz, Jiří Balhar, and David Mareček. 2023. Tokenization Impacts Multilingual Language Modeling: Assessing Vocabulary Allocation and Overlap Across Languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5661–5681, Toronto, Canada. Association for Computational Linguistics.

Tomasz Limisiewicz, Terra Blevins, Hila Gonen, Orevaoghene Ahia, and Luke Zettlemoyer. 2024. MYTE:

Morphology-Driven Byte Encoding for Better and Fairer Multilingual Language Modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15059–15076, Bangkok, Thailand. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, and 2 others. 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, and Yejin Choi. 2025. SuperBPE: Space Travel for Language Models. In *Proceedings of the Conference on Language Modeling (COLM)*.

Yikang Liu, Yeting Shen, Hongao Zhu, Lilong Xu, Zhiheng Qian, Siyuan Song, Kejia Zhang, Jialong Tang, Pei Zhang, Baosong Yang, Rui Wang, and Hai Hu. 2024. ZhoBLiMP: A Systematic Assessment of Language Models with Linguistic Minimal Pairs in Chinese. *Preprint*, arXiv:2411.06096.

Jonas F. Lotz, António V. Lopes, Stephan Peitz, Hendra Setiawan, and Leonardo Emili. 2025. Beyond Text Compression: Evaluating Tokenizers Across Scales. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32155–32173, Vienna, Austria. Association for Computational Linguistics.

Dominik Macháček, Jonáš Vidra, and Ondřej Bojar. 2018. Morphological and Language-Agnostic Word Segmentation for NMT. In *Text, Speech, and Dialogue*, pages 277–284, Cham. Springer International Publishing.

Clara Meister. 2026. TokEval: A Tokenizer Evaluation Suite. *Preprint*, arXiv:2608.18062.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual Minimal Pair Sentences for testing Robust Property Knowledge and its Inheritance in Pre-trained Language Models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Luca Moroni, Simone Conia, Federico Martelli, and Roberto Navigli. 2024. Towards a more comprehensive evaluation for Italian LLMs. In *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 584–599, Pisa, Italy. CEUR Workshop Proceedings.

Mosaic Research. 2023. Introducing MPT-7B: A New Standard for Open-Source, Commercially Usable LLMs. Accessed: 2026-01-28.

Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A. Raffel. 2023. Scaling Data-Constrained Language Models. In *Advances in Neural Information Processing Systems*, volume 36, pages 50358–50376. Curran Associates, Inc.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation. *Preprint*, arXiv:2207.04672.

Antoine Nzeyimana and Andre Niyongabo Rubungo. 2022. KinyaBERT: a Morphology-aware Kinyarwanda Language Model. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5347–5363, Dublin, Ireland. Association for Computational Linguistics.

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

Kyubyong Park, Joohong Lee, Seongbo Jang, and Dawoon Jung. 2020. An Empirical Study of Tokenization Strategies for Various Korean NLP Tasks. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 133–142, Suzhou, China. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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

of the Second Conference on Language Modeling, Montreal, Canada. ArXiv:2506.20920.

Aleksandar Petrov, Emanuele La Malfa, Philip H.S. Torr, and Adel Bibi. 2023. Language Model Tokenizers Introduce Unfairness Between Languages. In *Advances in Neural Information Processing Systems*, volume 36, pages 36963–36990.

Steven T. Piantadosi. 2014. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130.

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, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. OpenAI Blog.

Angelika Romanou, Negar Foroutan, Anna Sotnikova, Zeming Chen, Sree Harsha Nelaturu, Shivalika Singh, Rishabh Maheshwary, Micol Altomare, Mohamed A. Haggag, A. Snegha, Alfonso Amayuelas, Azril Hafizi Amirudin, Viraat Aryabumi, Danylo Boiko, Michael Chang, Jenny Chim, Gal Cohen, Aditya Kumar Dalmia, Abraham Diress, and 40 others. 2025. INCLUDE: Evaluating multilingual language understanding with regional knowledge. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740. AAAI Press.

Jonne Saleva and Constantine Lignos. 2021. The Effectiveness of Morphology-aware Segmentation in Low-Resource Neural Machine Translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 164–174, Online. Association for Computational Linguistics.

Craig W Schmidt, Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. 2024. Tokenization Is More Than Compression. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 678–702, Miami, Florida, USA. Association for Computational Linguistics.

Andrea Seveso, Daniele Potertì, Edoardo Federici, Mario Mezzanzanica, and Fabio Mercorio. 2025. ITALIC: An Italian Culture-Aware Natural Language Benchmark. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1469–1478, Albuquerque, New Mexico. Association for Computational Linguistics.

Amanda Silberling. 2024. Why AI Can't Spell 'Strawberry'. *TechCrunch*. Accessed: 2026-01-26.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, and 4 others. 2025. Global MMLU: Understanding and Addressing Cultural and Linguistic Biases in Multilingual Evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria. Association for Computational Linguistics.

Wenfang Sun, Xinyuan Song, Pengxiang Li, Lu Yin, Yefeng Zheng, and Shiwei Liu. 2025. The curse of depth in large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. 2024. Scaling Laws with Vocabulary: Larger Models Deserve Larger Vocabularies. In *Advances in Neural Information Processing Systems*, volume 37, pages 114147–114179. Curran Associates, Inc.

Alexey Tikhonov and Max Ryabinin. 2021. It's All in the Heads: Using Attention Heads as a Baseline for Cross-Lingual Transfer in Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3534–3546, Online. Association for Computational Linguistics.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinuç, 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, and 1 others. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Omri Uzan and Yuval Pinter. 2026. Charbench: Evaluating the role of tokenization in character-level tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33296–33304.

Mihaela Alexandra Vasiu and Rodica Potolea. 2020. Enhancing Tokenization by Embedding Romanian Language Specific Morphology. In *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 243–250.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Xilong Wang, Hao Fu, Jindong Wang, and Neil Zhenqiang Gong. 2025. StringLLM: Understanding the String Processing Capability of Large Language Models. In *The Thirteenth International Conference on Learning Representations (ICLR)*. ArXiv:2410.01208.

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

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Thilini Wijesiriwardene, Ruwan Wickramarachchi, Bimal Gajera, Shreeyash Gowaikar, Chandan Gupta, Aman Chadha, Aishwarya Naresh Reganti, Amit Sheth, and Amitava Das. 2023. ANALOGICAL - A Novel Benchmark for Long Text Analogy Evaluation in Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3534–3549, Toronto, Canada. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, and 13 others. 2020. CLUE: A Chinese Language Understanding Evaluation Benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online). International Committee on 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*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

George K. Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley.

Vilém Zouhar, Clara Meister, Juan Gastaldi, Li Du, Mrinmaya Sachan, and Ryan Cotterell. 2023. Tokenization and the Noiseless Channel. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5184–5207, Toronto, Canada. Association for Computational Linguistics.

A Intrinsic Tokenizer Metrics

We evaluate tokenizers using a comprehensive suite of intrinsic metrics spanning information-theoretic, linguistic, statistical, and cognitive dimensions. Unless otherwise specified, we use implementations from the `tokenizer-intrinsic-evals` (TokEval) suite (Meister, 2026), which we extended with additional metrics for this study. Metrics are computed on samples from the OSCAR 23.01 corpus (Abadji et al., 2022), a large-scale multilingual web corpus derived from Common Crawl. Table 2 provides a summary of all metrics computed.

A.1 Information Theoretic Metrics

Information-theoretic metrics have traditionally served as the primary tool for evaluating compression-based hypotheses of tokenizer intrinsic performance.

Rényi Entropy. The Rényi entropy of order α generalizes Shannon entropy and is defined as:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log_2 \left(\sum_i p_i^\alpha \right)$$

where X is the token distribution and p_i is the probability of token i. For $\alpha = 1$, this reduces to Shannon entropy. Higher-order Rényi entropy (e.g., $\alpha = 2$) penalizes token distributions that are too unbalanced, particularly those with very high-frequency tokens (Zouhar et al., 2023). We compute Rényi entropy for $\alpha \in \{1.0, 2.0, 2.5\}$.

Compression Ratio and Fertility. Compression ratio is the ratio of words to the number of tokens needed to represent a corpus:

$$\text{Compression Ratio} = \frac{\text{\# words}}{\text{\# tokens}}$$

Higher values indicate better compression. Fertility is the inverse: the average number of tokens produced per word. Lower fertility indicates more aggressive compression.

N-gram Entropy. Shannon entropy computed over token bigram (or higher-order n-gram) distributions:

$$H(X_n) = -\sum_{g \in \mathcal{G}_n} p(g) \log_2 p(g)$$

where X_n is the n-gram distribution, $\mathcal{G}_n$ is the set of all observed n-grams, and $p(g)$ is the empirical probability of n-gram g. Higher values indicate that tokens appear in more diverse contexts, which may correlate with more informative tokenization.

Unigram Distribution Entropy. Shannon entropy of the token frequency distribution, measuring the diversity of the unigram token distribution. Higher values indicate a more uniform usage of the vocabulary.

Encoding Length. The total number of tokens required to encode a corpus. This raw count provides an absolute measure of tokenization efficiency for a fixed text.

Token Length (Average Characters per Token). The average number of characters per token when encoding a given corpus. Longer tokens indicate more aggressive merging during vocabulary construction.

Average Tokens per Line. The mean number of tokens produced per line of text. This provides a coarse measure of tokenization granularity relative to natural text structure.

A.2 Zipfianness Metrics

Word frequency distributions in natural language follow Zipf's law (Zipf, 1949), which states that the frequency of a word is inversely proportional to its rank: $f(r) \propto r^{-\alpha}$ where r is the rank and $\alpha \approx 1$. This pattern holds across languages (Piantadosi, 2014). On a log-log plot, a perfect Zipfian distribution appears as a straight line with slope -1.

Recent work has shown that tokenizers whose frequency distributions adhere more closely to Zipf's law tend to yield better downstream model performance, particularly in multilingual settings (Lotz et al., 2025). The intuition is that tokenizers yielding token distributions closely aligned with a Zipfian power law may be better suited for modeling natural language. Following Lotz et al. (2025), we compute three metrics on the log-log rank-frequency curve. Following evidence that power laws only apply above some minimum rank (Clauset et al., 2009), we restrict our analysis to tokens with $\log(\text{rank}) \leq 6$.

Figure 4 illustrates token frequencies against frequency rank on log-log scales for English and three non-English languages. Rank-frequency patterns for English differ minimally across tokenizers, while non-Latin scripts like Russian and Chinese exhibit the largest distributional differences across tokenizers.

Rank-Frequency AUC. The area under the curve of sorted token frequency against frequency

Table 2: Summary of intrinsic tokenizer metrics computed in this study. Metrics marked with † were implemented in our fork of `tokenizer-intrinsic-evals`. Arrows indicate the direction proposed in the literature to correlate with better downstream performance; blank cells indicate metrics without a clear proposed direction.

Category	Metric		Description
Information Theoretic	`renyi_efficiency_a_*`	↑	Rényi entropy ($\alpha \in \{1, 2, 2.5\}$)
	`unigram_entropy`	↑	Shannon entropy of token distribution
	`compression_ratio`	↑	Normalization units per token
	`bigrams_entropy`	↑	Entropy over token bigrams
	`fertility`	↓	Tokens per normalization unit
	`encoding_length`†	↓	Total tokens to encode corpus
	`avg_chars_per_token`	↑	Average character length per token
	`avg_tokens_per_line`	↓	Average tokens per line of text
Zipfianness	`AUC_zipf`†		Area under log-log rank-frequency curve
	`SLOPE_zipf`†	≈ -1	Slope of linear fit on log-log
	`POWER_LAW_MAE_zipf`†	↓	Deviation from Zipf's law
Morphological	`morphscore_precision`	↑	Precision of morpheme boundary detection
	`morphscore_recall`	↑	Recall of morpheme boundary detection
	`morphscore_micro_f1`	↑	Micro-averaged F1 score
	`morphscore_macro_f1`	↑	Macro-averaged F1 score
Token Statistics	`vocab_size`		Size of tokenizer vocabulary
	`vocab_utilization`	↑	Proportion of vocabulary used
	`type_token_ratio`		Ratio of unique to total tokens
	`gini_coefficient`		Inequality of tokens-per-line across languages
	`final_tokens_count`†		Tokens not used in merges
	`used_in_merges`†		Tokens used as merge components
	`leading_space_tokens_*`†		Tokens with leading space marker
	`average_token_rank`		Mean frequency rank of tokens
Cognitive Plausibility	`corr_chunkability_rt_mean`†	↓	Correlation with reaction time
	`corr_chunkability_rt_z`†	↓	Correlation with z-scored RT
	`corr_chunkability_accuracy`†	↑	Correlation with accuracy
Boundary Statistics	`u_mean`†	↑	Mean U-statistic (PMI + entropy)
	`u_median`†	↑	Median U-statistic
	`u_mean_weighted`†	↑	Frequency-weighted mean U-statistic
	`u_mean_log_weighted`†	↑	Log-frequency-weighted mean U-statistic

rank in log-log scale, computed using Simpson's rule. This captures the overall shape of the rank-frequency distribution, summarizing how token frequencies decay with rank.

Slope (SLOPE). The slope β_1 from fitting a linear function $f(x) = \beta_0 + \beta_1 x$ on the log-log points (where $x = \log(\text{rank})$ and $y = \log(\text{count})$), approximating Zipf's law. For a perfect Zipfian distribution, the slope would be approximately -1. Deviations from -1 indicate that the tokenizer's frequency distribution differs from natural language patterns.

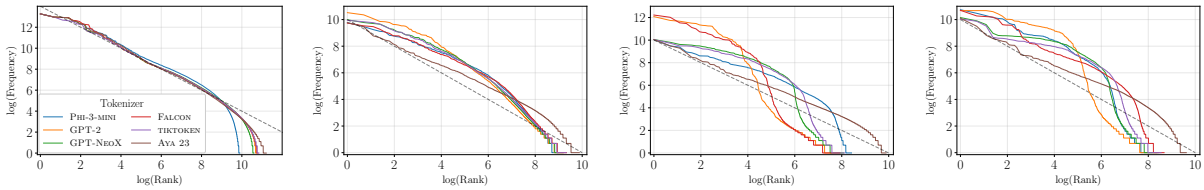
Power Law Deviation (POWER_LAW_MAE). The mean absolute error from the fitted linear function:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\beta_0 + \beta_1 x_i - y_i|$$

This metric quantifies how closely the token distribution aligns with a Zipfian power law. Lower values indicate closer adherence; high MAE suggests the token frequency distribution deviates substantially from a power law, which may indicate suboptimal vocabulary construction.

A.3 Linguistic and Morphological Metrics

MorphScore. Assesses the extent to which a tokenizer segments words along morpheme boundaries (Arnett and Bergen, 2025; Arnett et al., 2025b). If tokens follow morphological boundaries, they are more likely to be linguistically meaningful rather than arbitrary statistical artifacts. We report both precision (proportion of predicted boundaries that are correct) and recall (proportion of true boundaries that are recovered), as well as micro and macro F1 scores across languages.



(a) English (b) Czech (c) Russian (d) Chinese

Figure 4: Token frequency vs. frequency rank in log-log scale. Dashed lines (slope -1) reference Zipfian distributions. Figure from Lotz et al. (2025).

A.4 Token Statistics Metrics

We include metrics based on shallow token statistics that may capture phenomena complementary to information-theoretic measures.

Vocabulary Utilization. The proportion of the vocabulary actually used when encoding a corpus:

$$\text{Utilization} = \frac{\text{\# unique tokens used}}{\text{vocabulary size}}$$

Higher utilization suggests more efficient use of the available vocabulary.

Type-Token Ratio (TTR). The ratio of unique tokens (types) to total tokens in the corpus, measuring lexical diversity:

$$\text{TTR} = \frac{\text{\# unique tokens}}{\text{\# total tokens}}$$

Higher values indicate more diverse token usage.

Gini Coefficient (Tokenizer Fairness). A measure of how equitably a tokenizer distributes encoding costs across languages (Ahia et al., 2023; Petrov et al., 2023). For each language ℓ, the token cost c_ℓ is the number of tokens required per line of a content-matched corpus. The metric is the Gini coefficient of these per-language costs:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |c_i - c_j|}{2n^2 \mu}$$

where n is the number of languages and μ is the mean cost. Values range from 0 (all languages equally costly) upward, with higher values indicating that some languages require disproportionately more tokens than others for equivalent content. The metric is only defined across multiple languages; for the monolingual TIMTC and VISTA runs it is degenerate.

Average Token Rank. The expected rank of a token sampled from a corpus, where tokens are ranked by their frequency:

$$\text{AR} = \sum_{v \in \mathcal{V}} \text{rank}(v) \cdot p(v)$$

where $\mathcal{V}$ is the vocabulary, $\text{rank}(v)$ is the position of token v when vocabulary items are sorted by frequency (most frequent = rank 1), and $p(v) = \frac{f(v)}{\sum_{u \in \mathcal{V}} f(u)}$ is the empirical probability of token v in the corpus, with $f(v)$ being the frequency count of token v. Lower values indicate that the tokenizer relies more heavily on frequent tokens (Limisiewicz et al., 2023).

A.5 Cognitive Plausibility

Chunkability Correlation. Following Beinborn and Pinter (2023), we evaluate tokenizers through the lens of cognitive plausibility by correlating tokenizer behavior with human lexical decision performance. We use the British Lexicon Project (Keuleers et al., 2012), which contains lexical decision data (reaction time and accuracy) for 28,730 English words. Chunkability measures how aggressively a tokenizer segments a word:

$$\text{Chunkability}(w) = 1 - \frac{\text{\# tokens for word } w}{\text{\# characters in } w}$$

Chunkability ranges from 0 (each character is a separate token) to values approaching 1 (the word is kept as a single token). Higher chunkability indicates that the tokenizer treats the word as a more cohesive unit. We compute Pearson correlations between chunkability and three human performance measures: (1) mean reaction time (RT), the average time in milliseconds participants take to decide whether a letter string is a valid word—faster responses indicate easier processing; (2) z-scored reaction time, which normalizes RT within each participant to control for individual differences in baseline speed, providing a more reliable measure

of relative processing difficulty; and (3) accuracy percentage, the proportion of correct responses for each word.

Cognitively plausible tokenization should show positive correlation with accuracy (words kept intact are recognized more accurately) and negative correlation with reaction time (words kept intact are recognized faster). Prior work has shown that the Unigram algorithm yields less cognitively plausible tokenization behavior than BPE and WordPiece, and also exhibits worse coverage of derivational morphemes (Beinborn and Pinter, 2023).

A.6 Boundary Statistics (U-Statistics)

The U-statistic evaluates whether a token's boundaries align with natural linguistic boundaries by combining two signals from unsupervised segmentation: *internal cohesion* and *contextual separability* (Hu et al., 2025). The combined score for a token w is defined in Equation 1 (Section 3). Here we detail each component.

Internal cohesion: minimum adjacent-character PMI. For a token w consisting of characters $c_1, c_2, \ldots, c_n$, we compute the pointwise mutual information for each adjacent character pair:

$$\text{PMI}(c_i, c_{i+1}) = \log \frac{f(c_i c_{i+1}) \cdot T}{f(c_i) \cdot f(c_{i+1})}$$

where $f(\cdot)$ denotes corpus frequency and T is the total number of character bigrams in the corpus. High PMI means the two characters co-occur far more often than expected by chance, indicating they are part of the same linguistic unit (e.g., "t" and "h" in English, or "qu" in French).

We take the *minimum* PMI across all adjacent pairs within the token:

$$\text{PMI}_{\text{min}}(w) = \min_{(c_i, c_{i+1}) \subset w} \text{PMI}(c_i, c_{i+1})$$

This acts as a bottleneck: a single weak link—a character pair with low PMI—lowers the overall cohesion score, preventing loosely connected parts from being treated as a single unit (Hu et al., 2025). A token like "the" has uniformly high internal PMI, while a poorly segmented token like "e_of" would have a low-PMI pair at the boundary it straddles.

Contextual separability: branching entropy. The second component measures whether the token's edges coincide with natural segmentation points. We compute the Shannon entropy over the distribution of characters observed at each boundary:

$$H_{\text{left}}(w) = -\sum_{l} P(l \mid w) \log P(l \mid w)$$
$$H_{\text{right}}(w) = -\sum_{r} P(r \mid w) \log P(r \mid w)$$

where l and r index the set of distinct characters that can occur immediately to the left and right of w in the corpus, respectively.

The intuition comes from Harris (1955): at natural linguistic boundaries (e.g., between words or morphemes), many different characters can appear, leading to high entropy. Mid-morpheme, the set of possible continuations is constrained (e.g., after "qu" in English, "e", "i", "a" are likely but few others), yielding low entropy. A token whose boundaries both have high branching entropy is likely to start and end at natural break points. We take the minimum of H_{left} and H_{right}, so that a single side with low diversity keeps the token from being split prematurely.

Combining the signals and aggregation. The U-statistic combines internal cohesion and contextual separability as in Equation 1. The two terms are complementary: PMI captures whether the token's *interior* is coherent, while branching entropy captures whether the token's *edges* are well-placed. A high U-statistic requires both—the token must be internally cohesive *and* its boundaries must fall at positions with diverse continuations. The weighting $\lambda = 4.0$ balances the two terms, as PMI values span a wider range than branching entropy; we adopt the optimal value found by Hu et al. (2025) for Chinese word segmentation on the PKU benchmark. We compute U_{stat} for every token in the tokenizer's vocabulary using character statistics estimated from a corpus, then report the mean (`u_mean`) and median (`u_median`) across the vocabulary. We also report two frequency-weighted variants designed to dampen the relationship between U-statistic values and vocabulary size, by giving more weight to tokens the model encounters more often. Let $f(w)$ denote the corpus frequency of token w and let $\mathcal{V}$ be the set of vocabulary items with a valid U-statistic. The frequency-weighted mean is:

$$\texttt{u_mean_weighted} = \frac{\sum_{w \in \mathcal{V}} f(w) \cdot U_{\text{stat}}(w)}{\sum_{w \in \mathcal{V}} f(w)}$$

The log-frequency-weighted mean dampens the influence of extremely frequent tokens:

$$\texttt{u_mean_log_weighted} = \frac{\sum_{w\in\mathcal{V}} \log(1+f(w))\cdot U_{\text{stat}}(w)}{\sum_{w\in\mathcal{V}} \log(1+f(w))}$$

B Model Details

We use three model suites in this work: TIMTC, TokSuite, and VISTA. The TIMTC models from Schmidt et al. (2024) are released under the AI Pubs Open RAIL-M license.[9] The TokSuite models from Altıntaş et al. (2025) are released under the MIT license. Our VISTA models are released under the Apache 2.0 license.

B.1 TIMTC

The TIMTC (Tokenization Is More Than Compression) models from Schmidt et al. (2024) consist of 350M parameter decoder-only language models trained on the Pile corpus. The suite includes 54 models at this scale, plus additional 1.3B and 2.4B parameter models for select configurations. Models were trained with three vocabulary sizes (32,768, 40,960, and 49,152 tokens) across 18 different tokenizer configurations. Table 3 summarizes these conditions, which vary along four dimensions: The models use an MPT (MosaicML Pretrained Transformer; Mosaic Research, 2023) decoder-only architecture and were trained on the Pile corpus, with tokenizer vocabularies constructed over MiniPile, a 6GB subset of English-only data.

Vocabulary Construction (Vocab Constr) The algorithm used to build the vocabulary. BPE iteratively merges the most frequent byte pairs; Unigram starts from a large vocabulary and prunes tokens that minimally decrease corpus likelihood; WordPiece merges pairs based on pointwise mutual information; SaGe incorporates contextual information via a skip-gram objective; and PathPiece minimizes the total token count via shortest-path segmentation (with PathPieceL breaking ties by longest token and PathPieceR breaking ties randomly).

Initial Vocabulary (Init Voc) PathPiece and SaGe require a large starting vocabulary from which tokens are iteratively pruned. Options include a vocabulary pre-trained with BPE or Unigram, or the most frequent byte n-grams from the corpus. This column is empty for methods like BPE and WordPiece, which build vocabularies bottom-up by merging smaller units.

Pre-tokenization (Pre-tok) Rules that restrict token boundaries before vocabulary construction. "FirstSpace" requires spaces to appear only at the start of tokens; "SpaceDigit" treats each space and digit as its own token; "FirstSpDigit" combines both constraints; "None" allows unrestricted token boundaries.

Segmentation (Segment) The algorithm for splitting text into tokens at inference time. "Merge" applies BPE merge rules; "Likelihood" maximizes unigram probability; "Greedy" takes the longest matching token left-to-right; "PathPieceL/R" finds the minimum-token segmentation.

Table 3: Summary of 350M parameter model vocabulary construction methods, showing initialization, pre-tokenization, and segmentation strategies. All models have been trained with different vocab sizes, specifically 32,768, 40,960 and 49,152 tokens.

Vocab Constr	Init Voc	Pre-tok	Segment
PathPieceL	BPE Unigram n-gram n-gram	FirstSpace FirstSpace FirstSpDigit FirstSpace	PathPieceL
Unigram		FirstSpace	Likelihood Greedy PathPieceL
BPE		FirstSpace	Merge Greedy PathPieceL
WordPiece		FirstSpace	Greedy
SaGe	BPE n-gram Unigram n-gram	FirstSpace FirstSpace FirstSpace FirstSpDigit	Greedy
PathPieceR	n-gram	SpaceDigit FirstSpDigit None	PathPieceR

B.2 TokSuite

The TokSuite models from Altıntaş et al. (2025) are designed to isolate the impact of tokenizer choice on language model behavior. The suite consists of 14 models sharing identical architecture, training data, compute budget, and initialization, differing only in their tokenizer. Tokenizers are drawn from popular open-source models including GPT-2, BERT, Llama, Qwen, Gemma, Phi, Aya, BLOOM, XGLM, Tekken, GPT-4o, Comma, and TokenMonster, as well as the byte-level ByT5 tokenizer. This

[9] https://www.licenses.ai/ai-pubs-open-railm-vz1

controlled setup enables direct comparison of tokenizer effects on downstream performance across English and multilingual tasks.

B.3 VISTA Models

B.3.1 Model Architecture

We train two families of LLAMA 1 (Touvron et al., 2023) language models:

- **VISTA-130M** ($\sim$130M parameters, comparable to GPT-1 (Radford et al., 2018)): 12 layers, hidden size 768, 12 attention heads.
- **VISTA-350M** ($\sim$350M parameters): 24 layers, hidden size 1024, 16 attention heads.

Both architectures use SiLU activations, RoPE positional embeddings, a maximum sequence length of 1024, and RMS normalization (ϵ=10^{-6}). The embedding and language modeling head dimensions are adjusted to match each vocabulary size; all other hyperparameters are held fixed across tokenizers. Our implementation is built on the codebase of Sun et al. (2025) using the Hugging Face Transformers library (Wolf et al., 2020). Table 7 details the full model and training configurations for the 130M and 350M scales, respectively. Note that the parameter sizes for both families vary significantly with the vocabulary sizes.

B.3.2 Hyperparameters

For the 130M models, we closely follow the pretraining protocol from Chang et al. (2026). All 130M models are trained for 10 epochs on their respective BP-scaled monolingual training data, following evidence that multiple-epoch pre-training is beneficial for smaller datasets (Muennighoff et al., 2023). We use a global batch size of 512 and apply attention dropout (0.1) to mitigate overfitting, with gradient clipping disabled (following (Chang et al., 2026)).

For the 350M models, we train on substantially larger data: the English models are trained for 1 epoch on the Chinchilla-optimal 7 billion tokens from FineWeb (Penedo et al., 2024), while non-English models are trained on equivalent bytes of FineWeb-2 (Penedo et al., 2025) data scaled by the language-specific byte premium. The global batch size remains at 512, with gradient clipping enabled, and attention dropout removed. For fair comparison, all configurations within each model scale are held constant across languages and vocabulary sizes.

B.3.3 Data

VISTA 130M. We utilize monolingual datasets from Chang et al. (2026), covering five languages: English (`eng_latn`), Thai (`tha_thai`), Urdu (`urd_arab`), Amharic (`amh_ethi`), and Vietnamese (`vie_latn`). Training splits are defined by Byte Premium (BP) scaling, such that each language contributes an information-theoretically equivalent amount of data (approximately 1 GB $\times$ the Byte Premium). The BP multipliers are derived from Arnett et al. (2024a) and detailed in Table 6. Evaluation splits consist of the disjoint last (BP-scaled) 8,000 lines of each monolingual dataset.

VISTA 350M. At the 350M scale, we follow a Chinchilla-optimal data regime. English models are trained on 7 billion tokens from FineWeb's 10 Billion token sample (Penedo et al., 2024), yielding approximately 31 GiB of tokenized training data (for vocab size 32,768). For non-English languages, we source data from FineWeb-2 (Penedo et al., 2025) and define each language's training data target as:

$$\texttt{target_bytes} = \text{BP} \times \texttt{eng_ref_bytes}$$

where `eng_ref_bytes` is the byte size of the 7B-token, tokenized English training file, for that corresponding tokenizer type and vocabulary size. Table 4 details the data budget for the BPE tokenizer at vocabulary size 32,768 (this is representative; the same BP-scaling procedure is applied per tokenizer/vocabulary combination with respect to the corresponding English reference).

Not all languages have sufficient data in FineWeb-2 to meet the BP-scaled target. Amharic has only 2.69 GiB of raw text—less than 5% of the required $\sim$54 GiB (tokenized)—and was excluded from 350M training to avoid severe overfitting. Urdu has approximately 25% of the required data; we compensate with multi-epoch training (3–4$\times$ depending on tokenizer/vocabulary), which converged well empirically. Thai and Vietnamese have ample data in FineWeb-2 and are trained for a single epoch.

B.3.4 Compute Costs

All VISTA models are trained on 4 or 8 NVIDIA H100/A100 GPUs in `bfloat16` mixed precision.

VISTA 130M. Each 130M model is trained for 10 epochs on the BP-scaled datasets ($\sim$1–2.7 GB per language). Wall-clock time per run ranges between 2 and 3 hours ($\sim$8–12 GPU-hours). The full

130M experimental suite comprises 80 runs (5 languages $\times$ 2 tokenizers $\times$ 8 vocabulary sizes), for a total of 640–960 GPU-hours.

VISTA 350M. Each 350M model is trained for 1 epoch on the Chinchilla-optimal data (31–85 GiB tokenized per language), except Urdu which trains for 3–4 epochs to compensate for data scarcity. The full 350M suite comprises 16 runs (4 languages $\times$ 2 tokenizers $\times$ 2 vocabulary sizes), as detailed in Table 5.

B.3.5 Language Specific Tokenization

For both BPE and Unigram variants, tokenizers trained on the corresponding Arnett et al. (2025a) monolingual corpus with varying vocabulary sizes from Arnett et al. (2025a) are used. At the 350M scale, we reuse the same monolingual tokenizers for the FineWeb-2 languages (Thai, Urdu, Vietnamese), as for the 130M experiments (e.g., FineWeb-2 Thai data is tokenized with the same `tha_thai` tokenizer). All text data is pre-tokenized into integer IDs before training. Each line in the training file corresponds to a sequence of space-separated token integers. Consecutive non-empty lines from the source text are concatenated and packed to fill the maximum sequence length of 1024 tokens. We prepend a `CLS` token to each packed sequence and append a `SEP` token after every individual text segment within that sequence. This packing is performed offline with no padding tokens written to disk, ensuring the model sees a continuous stream of data at high throughput. For the 350M English models, the tokenized data is truncated to get the first 7 billion tokens. For non-English 350M models, the split is defined by the target byte count (`BP` $\times$ `eng_ref_bytes`), with the evaluation set taken from the disjoint tail of the tokenized corpus.

Table 4: FineWeb-2 data budget for VISTA 350M models (for the BPE, vocab = 32,768 case). The *BP-scaled target* is the Chinchilla-optimal tokenized data size scaled by each language's byte premium. *Tokenized Data Size* is the actual tokenized data after processing downloaded FineWeb-2 text using the pre-trained language specific tokenizers.

	English	**Thai**	**Urdu**	**Amharic**[†]	**Vietnamese**
Byte Premium	1.00	2.74	1.71	1.72	1.35
English reference (7B tokens)			31.11 GiB		
BP-scaled target	31.11 GiB	85.25 GiB	53.20 GiB	53.52 GiB	42.00 GiB
FineWeb2 Raw Available	–	321.88 GiB	22.48 GiB	2.69 GiB	402.69 GiB
Tokenized Data Size	31.11 GiB	>85 GiB	12.54 GiB	–	38.27 GiB
Coverage	100%	>100%	∼24%	<5%	∼91%
Compute-matched Training epochs	1	1	3–4[‡]	∼20×[†]	1

[†]Amharic was excluded from 350M training: <5% data coverage would require ∼20× epoch repetition, causing severe overfitting. [‡]Urdu uses 3–4× epoch compensation across configs, which converged well empirically.

Table 5: VISTA Pre-training Variations.

	VISTA-130M	**VISTA-350M**
Languages	English, Thai, Urdu, Amharic, Vietnamese	English, Thai, Urdu, Vietnamese
Tokenizers	BPE, Unigram	BPE, Unigram
Vocabulary sizes	8192, 16384, 32768, 49152, 65536, 81920, 98304, 114688	8192, 98304
Total runs	$5 \times 2 \times 8 = \mathbf{80}$	$4 \times 2 \times 2 = \mathbf{16}$

Table 6: Monolingual corpora and byte-premium (BP) statistics used in our experiments. The left column shows the BP-scaled data size (in GB), and the right column shows the raw UTF-8 byte size. BP multipliers are derived from Arnett et al. (2024a); BP scaling ensures languages with less information-dense scripts are allocated proportionately more raw text data.

Language (tag)	**BP-scaled GB**	**Raw GB**
English (eng_latn)	1	1.00
Thai (tha_thai)	1	2.74
Urdu (urd_arab)	1	1.71
Amharic (amh_ethi)	1	1.72
Vietnamese (vie_latn)	1	1.35

Table 7: Model architecture and training hyperparameters for VISTA-130M and VISTA-350M. Where the two scales differ, both values are shown; shared values appear once spanning both columns.

Configuration	**VISTA-130M**	**VISTA-350M**
Model Architecture		
Model Type	LLaMA 1	
Hidden Size (d_{model})	768	1024
Intermediate Size (d_{ff})	2048	2736
Number of Hidden Layers	12	24
Number of Attention Heads	12	16
Max Sequence Length	1024	
Vocabulary Size	*Varying*	
Activation Function	SiLU	
RMS Norm Epsilon	1×10^{-6}	
Initializer Range	0.02	
Positional Embeddings	RoPE	
Training Hyperparameters		
Learning Rate (Max)	1×10^{-3}	
Learning Rate (Min)	1×10^{-4} (min_lr_ratio = 0.1)	
LR Scheduler	Cosine with Warmup	
Warmup Steps Ratio	0.10	
Attention Dropout	0.1	0.0 (Disabled)
Adam β_1, β_2	0.9, 0.999	
Adam Epsilon	1×10^{-6}	
Weight Decay	0.0	
Precision	bfloat16	
Gradient Clipping	0.0 (Disabled)	1.0
Epochs	10	1*
Batch Configuration		
World Size	4 GPUs (H100/A100)	
Per-GPU Micro-batch Size	32	16
Gradient Accumulation Steps	4	8
Global Batch Size	512	
Data		
Training data source	Goldfish (Chang et al., 2026)	FineWeb/FineWeb-2
Training data size	$\sim$1 GB $\times$ BP	7B tokens (Eng); BP $\times$ Eng ref bytes

*Urdu: 3–4 epochs to compensate for limited FineWeb-2 data ($\sim$24% of BP-scaled target).

C Eval Dataset Details

We list the benchmarks we used to evaluate each model suite: TokSuite (Table 8), TIMTC (Table 9), and VISTA (Table 10). All benchmarks are released under permissive open-source licenses, suitable for this work. FLORES, Global PIQA, ARC, and Belebele are released under CC-BY-SA-4.0 licenses. BLiMP, MultiBLiMP, Winograd_th, XWinograd, and XCOPA are released under CC-BY-4.0 licenses. HellaSwag and MultiLoko are released under MIT licenses. Global MMLU, AfriMMLU, and INCLUDE are released under Apache-2.0 licenses.

Table 8: List of language model evaluations used for TokSuite models. Tasks marked with † were implemented in our fork of the LM evaluation harness.

Task Type	Benchmark	Language(s)
Linguistic	CoLA (Warstadt et al., 2019)	English
	BLiMP (Warstadt et al., 2020)	English
	ZhoBLiMP (Liu et al., 2024)	Chinese
	TurBLiMP (Başar et al., 2025)	Turkish
	MultiBLiMP (Jumelet et al., 2026)	English, Italian, Farsi, Turkish
NLU	SuperGLUE (Wang et al., 2019)	English
	Winogrande (Sakaguchi et al., 2020)	English, Turkish
	ANALOGICAL † (Wijesiriwardene et al., 2023)	English
	XWinograd (Tikhonov and Ryabinin, 2021)	Chinese
	CLUE † (Xu et al., 2020)	Chinese
	FarsTail † (Amirkhani et al., 2020)	Farsi
	Turkish NLI † (Budur et al., 2020)	Turkish
Commonsense Reasoning	HellaSwag (Zellers et al., 2019)	English
	PIQA (Bisk et al., 2020)	English
	PROST (Aroca-Ouellette et al., 2021)	English
	XCOPA (Ponti et al., 2020)	Chinese, Turkish
	Global PIQA (Chang et al., 2025)	Italian, Turkish, Chinese
	XStoryCloze (Lin et al., 2022)	Chinese
World Knowledge/ QA	COMPS † (Misra et al., 2023)	English
	MMLU (Hendrycks et al., 2021)	English
	EWoK † (Ivanova et al., 2025)	English
	XCOMPS † (He et al., 2025)	Farsi, Turkish, Chinese
	XQuAD (Artetxe et al., 2020)	English, Turkish, Chinese
	CMMLU (Li et al., 2024)	Chinese
	Global MMLU (Singh et al., 2025)	English, Italian, Farsi, Turkish, Chinese
	Belebele (Bandarkar et al., 2024)	English, Italian, Farsi, Turkish, Chinese
	INCLUDE (Romanou et al., 2025)	Chinese, Farsi, Turkish
	MultiLoKo † (Hupkes and Bogoychev, 2025)	Farsi, Chinese, Turkish
	PersianQA † (Ayoubi and Davoodeh, 2021)	Farsi
	SynTran-fa † (Farsi et al., 2024)	Farsi
Multi-Task	BBH (Suzgun et al., 2023)	English
	ITALIC † (Seveso et al., 2025)	Italian
	ITA-Bench † (Moroni et al., 2024)	Italian
Token-Level	CUTE † (Edman et al., 2024)	English
	CharBench † (Uzan and Pinter, 2026)	English
	LMentry † (Efrat et al., 2023)	English
	stringbench † (Wang et al., 2025)	English
	EXECUTE † (Edman et al., 2025)	English, Chinese
	TokSuite (Altıntaş et al., 2025)	English, Chinese, Farsi, Italian
Perplexity	FLORES † (NLLB Team et al., 2022)	English, Italian, Farsi, Turkish, Chinese

Table 9: List of English language model evaluations used for "Tokenization is More Than Compression" models. Tasks marked with † were implemented in our fork of the LM evaluation harness.

Task Type	Benchmark
Linguistic	SuperGLUE (Wang et al., 2019) Winogrande (Sakaguchi et al., 2020) BLiMP (Warstadt et al., 2020) MultiBLiMP (Jumelet et al., 2026)
World Knowledge/ Semantics	COMPS † (Misra et al., 2023) ANALOGICAL † (Wijesiriwardene et al., 2023) EWoK † (Ivanova et al., 2025) MMLU (Hendrycks et al., 2021) Global MMLU (Singh et al., 2025)
Commonsense Reasoning	PIQA (Bisk et al., 2020) HellaSwag (Zellers et al., 2019) PROST (Aroca-Ouellette et al., 2021)
QA / Reading	XQuAD (Artetxe et al., 2020) Belebele (Bandarkar et al., 2024)
Multi-Task	BBH (Suzgun et al., 2023)
Token-Level	CUTE † (Edman et al., 2024) EXECUTE † (Edman et al., 2025) CharBench † (Uzan and Pinter, 2026) stringbench † (Wang et al., 2025) LMentry † (Efrat et al., 2023) TokSuite (Altıntaş et al., 2025)
Perplexity	FLORES † (NLLB Team et al., 2022)

Table 10: List of language model evaluations used for VISTA. Tasks marked with † were implemented in our fork of LM evaluation harness.

Task Type	Benchmark	Language(s)
Perplexity	FLORES † (NLLB Team et al., 2022)	Amharic, English, Thai, Urdu, Vietnamese
Linguistic	BLiMP (Warstadt et al., 2020)	English
	MultiBLiMP (Jumelet et al., 2026)	Amharic, English, Urdu
NLU	SuperGLUE (Wang et al., 2019)	English
	Winograd_th †	Thai
	Winogrande (Sakaguchi et al., 2020)	English
	XNLI (Conneau et al., 2018)	Vietnamese
Commonsense Reasoning	HellaSwag (Zellers et al., 2019)	English, Vietnamese
	PIQA (Bisk et al., 2020)	English
	PROST (Aroca-Ouellette et al., 2021)	English
	Global PIQA (Chang et al., 2025)	Amharic, English, Thai, Urdu, Vietnamese
	XCOPA (Ponti et al., 2020)	Thai, Vietnamese
	ARC (Clark et al., 2018)	Vietnamese
World Knowledge/ QA	COMPS † (Misra et al., 2023)	English
	ANALOGICAL † (Wijesiriwardene et al., 2023)	English
	EWoK † (Ivanova et al., 2025)	English
	MMLU (Hendrycks et al., 2021)	English
	XQuAD (Artetxe et al., 2020)	English, Thai, Vietnamese
	Global MMLU (Singh et al., 2025)	Amharic, English, Vietnamese
	MMLU Prox †	Thai, Urdu
	AfriMMLU †	Amharic
	Belebele (Bandarkar et al., 2024)	Amharic, English, Thai, Urdu, Vietnamese
	MultiLoKo † (Hupkes and Bogoychev, 2025)	Thai, Urdu, Vietnamese
	INCLUDE (Romanou et al., 2025)	Urdu, Vietnamese
	Amharic QA †	Amharic
	ViQuAD †	Vietnamese
	VMLU †	Vietnamese
Multi-Task	BBH (Suzgun et al., 2023)	English
Token-Level	CUTE † (Edman et al., 2024)	English
	EXECUTE † (Edman et al., 2025)	English
	CharBench † (Uzan and Pinter, 2026)	English
	stringbench † (Wang et al., 2025)	English
	LMentry † (Efrat et al., 2023)	English
	TokSuite (Altıntaş et al., 2025)	English

D Full Eval Results

D.1 TokSuite

The evaluation results of the TokSuite models for each of the languages the models are trained on: English (Table 11), Chinese (Table 12), Turkish (Table 13), Farsi (Table 14), and Italian (Table 15).

Fig. 5 shows the model performance for each of the TokSuite models by language and macro averaged across language.

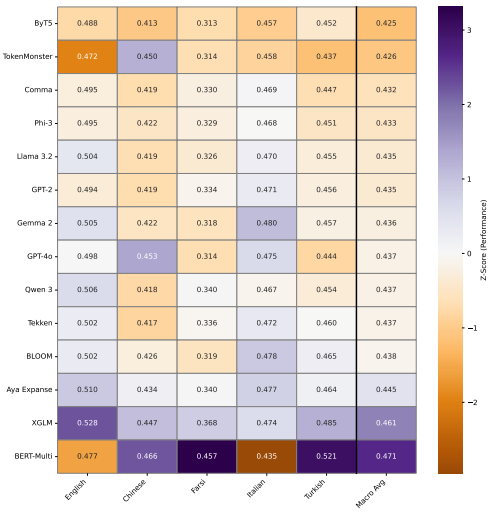


Figure 5: Downstream performance for TokSuite models aggregated by language and macro average across language. Each cell value shows mean performance. The color of each cell corresponds to the z-scored performance, where the darker the green the better the performance.

Table 11: Evaluation scores for TokSuite models on English tasks.

Tokenizer	Aya	Qwen	BLOOM	Comma	XGLM	BERT	ByT5	Gemma	GPT-2	Llama	Phi	Tekken	GPT-4o	TokenMonster	Chance
GLUE	0.448	0.446	0.420	0.430	0.397	0.402	0.394	**0.459**	0.418	0.394	0.414	0.451	0.429	0.418	0.45
Winogrande	0.584	0.579	0.586	0.560	0.577	0.525	0.553	0.568	0.542	**0.593**	0.586	0.576	0.582	0.539	0.50
BLiMP	0.835	0.836	0.833	0.843	0.835	0.833	0.838	0.838	0.827	**0.844**	0.836	0.837	**0.844**	0.787	0.50
MultiBLiMP	0.990	0.992	0.992	0.984	0.988	**0.994**	0.990	0.988	0.990	0.988	**0.994**	0.992	0.990	0.982	0.50
EWoK	0.629	0.623	0.622	0.608	0.615	0.563	0.619	0.627	0.615	**0.632**	0.614	0.614	0.623	0.600	0.50
PIQA	0.729	0.724	0.726	0.722	0.726	0.669	0.696	**0.739**	0.716	0.724	0.711	0.712	0.721	0.702	0.50
HellaSwag	0.437	0.427	0.426	0.412	**0.441**	0.408	0.403	0.435	0.412	0.430	0.414	0.427	0.431	0.387	0.25
PROST	**0.285**	0.282	0.260	0.229	0.228	0.253	0.250	0.236	0.244	0.264	0.244	0.256	0.262	0.214	0.25
MMLU	0.236	0.257	0.234	0.242	0.246	0.229	0.235	0.253	**0.260**	0.251	0.252	0.240	0.246	0.235	0.25
BBH	0.233	0.225	0.236	0.228	0.200	0.203	0.210	0.211	**0.242**	0.237	0.240	0.234	0.223	0.173	0.33
XQuAD	**0.105**	0.056	0.082	0.065	0.000	0.000	0.025	**0.105**	0.048	0.075	0.064	0.056	0.013	0.056	0.00
Belebele	0.402	0.407	0.386	0.394	**0.422**	0.288	0.390	0.416	0.392	0.389	0.378	0.398	0.401	0.378	0.25
COMPS	0.682	0.670	0.674	0.651	0.679	0.679	**0.689**	0.660	0.669	0.679	0.672	0.660	0.673	0.668	0.50
ANALOGICAL	0.816	0.816	0.810	0.809	0.789	0.359	0.781	0.792	0.768	0.799	0.751	**0.821**	0.794	0.699	0.25
Avg.	**0.502**	0.498	0.486	0.480	0.475	0.395	0.465	0.488	0.470	0.487	0.473	0.485	0.482	0.461	0.38

Table 12: Evaluation scores for TokSuite models on Chinese tasks.

Tokenizer	Aya	Qwen	BLOOM	Comma	XGLM	BERT	ByT5	Gemma	GPT-2	Llama	Phi	Tekken	GPT-4o	TokenMonster	Chance
ZhoBLiMP	0.872	0.872	0.876	0.879	0.871	0.871	0.860	0.876	0.879	0.879	0.882	0.874	0.871	**0.891**	0.50
XWinograd	0.724	0.708	0.690	0.738	0.712	0.687	0.716	0.732	0.730	0.722	0.732	0.690	**0.744**	0.702	0.50
XCOMPS	0.667	0.675	**0.703**	0.670	0.668	0.702	0.686	0.635	0.686	0.665	0.690	0.676	0.667	0.690	0.50
XCOPA	0.642	0.634	0.630	0.620	0.624	0.610	0.596	0.644	0.616	**0.648**	0.634	0.644	0.642	0.624	0.50
XStoryCloze	0.592	0.592	0.595	0.578	**0.600**	0.567	0.561	0.592	0.567	0.587	0.572	0.590	0.584	0.568	0.50
XQuAD	**0.057**	0.007	0.034	0.006	0.000	0.000	0.051	0.013	0.035	0.020	0.011	0.015	0.000	0.003	0.00
Global PIQA	0.510	0.520	**0.560**	0.540	0.510	0.480	0.510	0.500	0.480	0.480	0.540	0.480	0.540	0.540	0.50
Belebele	0.383	0.337	0.358	0.364	**0.386**	0.327	0.360	0.361	0.359	0.368	0.379	0.356	0.376	0.368	0.25
Global MMLU	0.273	0.250	0.253	0.260	0.223	0.225	0.263	0.253	**0.275**	0.265	0.242	0.230	0.260	**0.275**	0.25
CMMLU	0.247	0.251	0.245	0.249	0.249	**0.253**	0.251	0.250	0.249	0.246	0.249	0.250	0.247	0.247	0.25
MultiLoKo	**0.116**	0.064	0.052	0.012	0.004	0.000	0.008	0.096	0.020	0.028	0.044	0.072	0.020	0.000	0.00
INCLUDE	**0.290**	0.229	0.250	0.229	0.253	0.246	0.224	0.244	0.259	0.250	0.226	0.250	0.224	0.240	0.25
CLUE	0.419	0.435	0.438	**0.441**	0.439	0.345	0.435	0.434	0.435	0.435	0.435	0.432	0.435	0.435	0.39
Avg.	**0.472**	0.447	0.456	0.459	0.463	0.373	0.458	0.464	0.460	0.454	0.469	0.459	0.449	0.456	0.34

Table 13: Evaluation scores for TokSuite models on Turkish tasks.

Tokenizer	Aya	Qwen	BLOOM	Comma	XGLM	BERT	ByT5	Gemma	GPT-2	Llama	Phi	Tekken	GPT-4o	TokenMonster	Chance
Winogrande	0.533	0.519	**0.547**	0.488	0.509	0.483	0.516	0.511	0.506	0.500	0.516	0.501	0.501	0.502	0.50
MultiBLiMP	0.959	0.950	0.958	**0.964**	0.962	0.961	0.960	0.954	0.952	0.943	0.945	0.947	0.949	0.959	0.50
TurBLiMP	0.831	0.840	0.827	0.840	0.836	**0.843**	0.831	0.835	0.841	0.835	0.833	0.835	0.829	0.805	0.50
XCOMPS	0.656	0.662	0.651	0.654	**0.680**	0.675	0.655	0.644	0.662	0.666	0.641	0.662	0.660	0.612	0.50
XCOPA	0.632	**0.642**	0.610	0.614	0.638	0.596	0.616	0.622	0.636	**0.642**	0.610	0.630	0.626	0.622	0.50
Global PIQA	0.510	**0.550**	0.530	0.510	0.480	0.500	0.520	0.530	0.530	0.470	0.520	0.520	0.530	0.510	0.50
Belebele	0.371	0.364	0.357	0.357	0.373	0.273	0.363	0.361	0.368	**0.383**	0.349	**0.383**	0.376	0.354	0.25
MultiLoKo	0.124	0.060	0.068	0.080	0.016	0.000	0.072	0.064	0.092	0.092	0.100	**0.132**	0.008	0.008	0.00
INCLUDE	0.261	0.248	**0.308**	0.226	0.226	0.290	0.277	0.252	0.241	0.272	0.254	0.254	0.223	0.224	0.25
XQuAD	0.076	0.025	**0.092**	0.027	0.000	0.000	0.016	0.087	0.029	0.066	0.048	0.037	0.006	0.043	0.00
NLI	0.342	0.317	**0.358**	0.332	0.342	0.317	0.341	0.342	0.346	0.314	0.325	0.347	0.350	0.344	0.33
Avg.	0.481	0.471	**0.482**	0.463	0.460	0.449	0.470	0.473	0.473	0.471	0.467	0.477	0.460	0.453	0.35

D.1.1 Correlation Between Vocab Size and Performance

Figure 6 plots the relationship between vocabulary size (log-scaled; x-axis) and aggregate performance (mean for each subset, and macro average across language; y-axis) for the TokSuite models. For each facet, Spearman's ρ and p-value are reported.

Table 14: Evaluation scores for TokSuite models on Farsi tasks.

Tokenizer	Aya	Qwen	BLOOM	Comma	XGLM	BERT	ByT5	Gemma	GPT-2	Llama	Phi	Tekken	GPT-4o	TokenMonster	Chance
MultiBLiMP	0.930	0.941	0.929	0.948	**0.974**	0.968	0.926	0.876	0.944	0.899	0.942	0.908	0.900	0.946	0.50
XCOMPS	0.659	0.665	0.628	0.643	**0.677**	0.674	0.660	0.653	0.663	0.656	0.655	0.655	0.655	0.673	0.50
FarsTail	0.338	0.336	0.331	0.347	0.348	0.332	0.337	0.325	0.333	0.336	0.325	**0.362**	0.328	0.332	0.33
MultiLoKo	**0.136**	0.128	0.064	0.076	0.024	0.000	0.008	0.040	0.080	0.096	0.108	0.124	0.008	0.004	0.00
INCLUDE	0.223	**0.276**	0.217	0.268	0.241	0.221	0.230	**0.276**	**0.276**	0.246	0.261	0.228	**0.276**	0.241	0.25
SynTran	**0.018**	0.013	0.007	0.007	0.000	0.000	0.008	0.006	0.006	0.007	0.010	0.012	0.007	0.002	0.00
Persian QA	**0.114**	0.069	0.051	0.038	0.021	0.000	0.026	0.049	0.068	0.044	0.026	0.086	0.002	0.003	0.00
Belebele	0.376	0.366	0.373	0.381	**0.388**	0.282	0.366	0.369	0.374	0.379	0.370	0.384	0.378	0.362	0.25
Avg.	**0.397**	0.391	0.372	0.379	0.380	0.309	0.365	0.371	0.385	0.377	0.386	0.390	0.367	0.357	0.23

Table 15: Evaluation scores for TokSuite models on Italian tasks.

Tokenizer	Aya	Qwen	BLOOM	Comma	XGLM	BERT	ByT5	Gemma	GPT-2	Llama	Phi	Tekken	GPT-4o	TokenMonster	Chance
ITA-Bench	0.329	0.318	0.327	0.323	**0.338**	0.307	0.308	**0.338**	0.322	0.334	0.318	0.316	0.324	0.316	0.35
ARC Challenge	0.283	0.277	0.284	0.292	0.284	0.245	0.257	0.289	0.266	0.285	0.285	**0.297**	**0.297**	0.260	0.25
ARC Easy	0.560	0.544	0.547	0.542	0.564	0.385	0.538	0.564	0.538	0.556	0.509	0.554	**0.567**	0.523	0.25
HellaSwag	0.395	0.390	0.393	0.378	0.396	0.375	0.373	**0.399**	0.382	0.393	0.381	0.394	0.394	0.357	0.25
PIQA	0.654	0.652	0.639	0.637	0.654	0.633	0.619	0.642	0.642	0.640	0.631	**0.657**	0.650	0.633	0.50
Winogrande	0.610	0.604	0.609	0.599	0.606	0.553	**0.613**	0.612	0.607	0.605	0.606	0.593	0.601	0.581	0.50
ITALIC	0.295	0.281	0.294	0.287	0.288	0.266	0.283	**0.297**	0.294	0.288	**0.297**	0.287	0.290	0.278	0.25
MultiBLiMP	0.979	0.977	0.976	0.978	0.974	0.981	0.979	0.981	0.979	0.979	0.976	0.977	**0.983**	0.973	0.50
Belebele	0.368	0.354	0.369	0.361	**0.377**	0.297	0.366	**0.377**	0.366	0.370	0.343	0.363	0.368	0.354	0.25
Global MMLU	0.225	0.217	**0.273**	0.237	0.230	0.235	0.228	0.250	0.263	0.268	0.240	0.258	0.240	0.210	0.25
Avg.	0.509	0.501	0.508	0.496	0.512	0.409	0.492	0.516	0.505	**0.518**	0.501	0.497	0.508	0.492	0.37

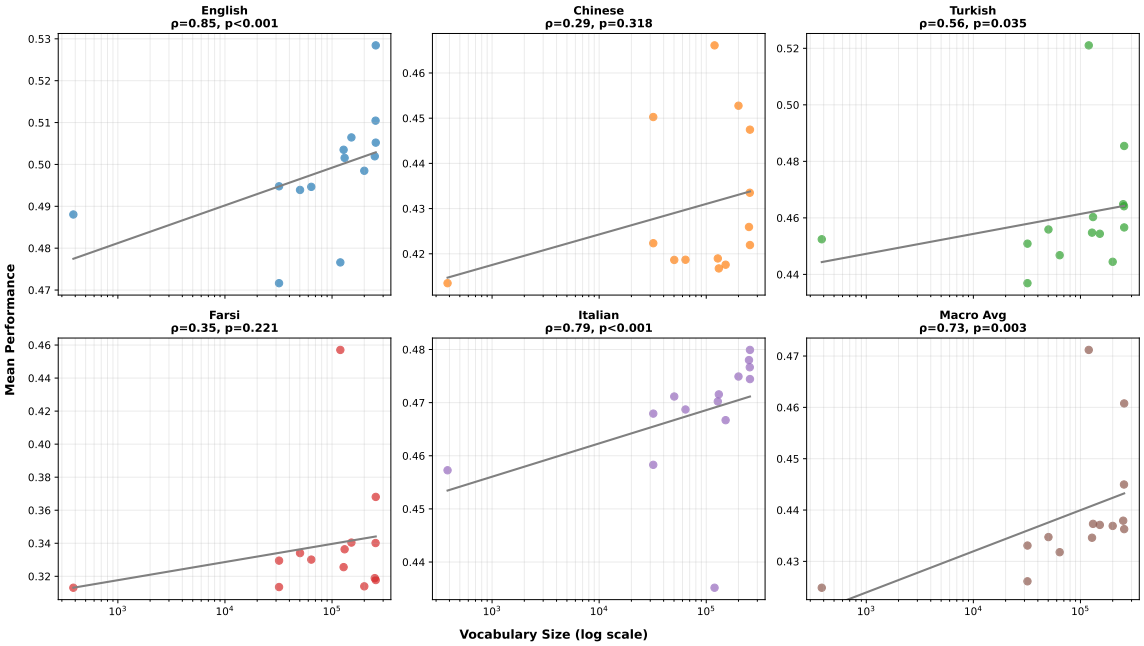


Figure 6: Correlation between vocab size and model performance for the TokSuite models. Facets represent evaluation language.

D.2 TIMTC

The evaluation results for the TIMTC models are shown in Tables 16–18, one per vocabulary size. Column labels abbreviate the tokenizer configurations described in Table 3.

Table 16: Evaluation scores for TIMTC models (vocab size 32,768) on English tasks.

	BPE	PPL-ng-fs	PPL-ng-fsd	PPL-uni-fs	PPR-fsd	PPR-none	PPR-sd	SAGE-bpe	SAGE-ng-fs	SAGE-ng-fsd	SAGE-uni	UNI-gr	UNI-lk	UNI-ppl	WP	Chance
GLUE	0.478	0.452	0.475	0.463	0.457	0.468	0.480	0.461	**0.482**	0.458	0.451	0.471	0.462	0.463	0.466	0.45
Winogrande	0.505	0.507	0.507	0.507	0.529	0.507	0.515	0.522	0.530	**0.541**	0.515	0.507	0.524	0.522	0.521	0.50
BLiMP	0.824	0.711	0.730	0.757	0.741	0.758	0.790	0.817	0.811	0.799	0.814	0.822	0.823	0.739	**0.829**	0.50
MultiBLiMP	0.983	0.916	0.919	0.899	0.917	0.951	0.983	**0.988**	0.986	0.981	0.975	0.987	0.981	0.887	0.979	0.50
EWoK	0.569	0.518	0.519	0.538	0.515	0.525	0.556	**0.570**	0.565	0.564	0.569	0.564	0.568	0.535	0.568	0.50
PIQA	0.653	0.607	0.611	0.616	0.607	0.612	0.641	**0.658**	0.653	0.657	0.655	0.657	**0.658**	0.614	0.649	0.50
HellaSwag	**0.327**	0.294	0.292	0.296	0.296	0.306	0.320	0.320	0.322	0.320	0.325	0.325	0.325	0.301	0.324	0.25
PROST	0.192	0.223	0.227	**0.248**	0.229	**0.248**	0.213	0.211	0.230	0.235	0.222	0.230	0.228	0.243	0.219	0.25
MMLU	0.233	**0.248**	0.240	0.234	0.240	0.240	0.235	0.239	0.240	0.230	0.231	0.233	0.232	0.231	0.231	0.25
BBH	0.003	0.000	0.000	0.000	0.000	**0.006**	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.33
XQuAD	0.000	0.000	0.001	0.000	0.000	0.000	0.001	0.000	0.000	0.000	**0.006**	0.000	0.000	0.004	0.000	0.00
Belebele	0.339	0.329	0.327	0.320	0.340	0.287	0.340	0.333	0.342	0.343	0.353	0.342	0.337	0.330	**0.359**	0.25
COMPS	**0.601**	0.520	0.527	0.558	0.519	0.560	0.564	0.589	0.588	0.582	0.595	0.582	0.598	0.552	**0.601**	0.50
ANALOGICAL	**0.647**	0.445	0.427	0.429	0.449	0.413	0.609	0.627	0.563	0.585	0.582	0.585	0.572	0.382	0.610	0.25
Avg.	**0.454**	0.412	0.414	0.419	0.417	0.420	0.446	0.453	0.451	0.450	0.450	0.450	0.451	0.415	**0.454**	0.36

Table 17: Evaluation scores for TIMTC models (vocab size 40,960) on English tasks.

	BPE	PPL-ng-fs	PPL-ng-fsd	PPL-uni-fs	PPR-fsd	PPR-none	PPR-sd	SAGE-bpe	SAGE-ng-fs	SAGE-ng-fsd	SAGE-uni	UNI-gr	UNI-lk	UNI-ppl	WP	Chance
GLUE	0.474	0.456	0.458	0.456	0.466	0.458	**0.479**	0.462	**0.478**	0.469	0.459	0.455	0.470	0.455	0.460	0.45
Winogrande	**0.536**	0.507	0.497	0.507	0.527	0.513	0.488	0.522	0.503	0.507	0.517	0.524	0.496	0.496	0.520	0.50
BLiMP	**0.825**	0.750	0.732	0.748	0.736	0.748	0.738	0.819	0.802	0.793	0.814	0.821	0.822	0.735	0.822	0.50
MultiBLiMP	0.977	0.919	0.922	0.865	0.918	0.951	0.927	**0.990**	0.984	0.974	0.984	0.983	0.982	0.869	0.982	0.50
EWoK	0.570	0.518	0.524	0.533	0.527	0.536	0.530	0.575	0.574	0.557	0.560	0.572	0.571	0.530	**0.577**	0.50
PIQA	0.659	0.602	0.604	0.615	0.602	0.609	0.600	**0.668**	0.659	0.645	0.657	0.655	0.655	0.608	0.665	0.50
HellaSwag	0.324	0.294	0.295	0.296	0.290	0.303	0.297	0.324	0.322	0.315	**0.327**	0.325	0.326	0.299	**0.327**	0.25
PROST	0.189	0.238	0.203	0.220	0.236	**0.252**	0.232	0.224	0.234	0.233	0.239	0.195	0.231	0.240	0.220	0.25
MMLU	0.240	0.229	0.235	**0.248**	0.229	0.243	0.231	0.229	0.247	0.231	0.231	0.231	0.231	0.236	0.241	0.25
BBH	0.002	**0.005**	0.002	0.000	0.000	0.002	0.000	0.002	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.33
XQuAD	0.000	0.000	0.000	0.002	0.000	0.000	0.001	0.000	0.000	0.001	**0.005**	0.000	0.000	0.000	0.000	0.00
Belebele	0.337	0.333	0.324	0.312	0.330	0.286	0.339	0.346	0.348	**0.360**	0.357	0.334	0.333	0.328	0.349	0.25
COMPS	0.603	0.515	0.519	0.557	0.519	0.563	0.522	**0.604**	0.578	0.552	0.592	0.588	**0.604**	0.555	0.599	0.50
ANALOGICAL	0.659	0.456	0.468	0.424	0.502	0.387	0.470	0.651	0.569	0.501	0.588	0.566	0.596	0.409	**0.660**	0.25
Avg.	0.457	0.416	0.413	0.413	0.420	0.418	0.418	**0.458**	0.450	0.439	0.452	0.446	0.451	0.411	**0.459**	0.36

Table 18: Evaluation scores for TIMTC models (vocab size 49,152) on English tasks.

	BPE	PPL-ng-fs	PPL-ng-fsd	PPL-uni-fs	PPR-fsd	PPR-none	PPR-sd	SAGE-bpe	SAGE-ng-fs	SAGE-ng-fsd	SAGE-uni	UNI-gr	UNI-lk	UNI-ppl	WP	Chance
GLUE	**0.474**	0.455	0.455	0.463	0.461	0.459	0.456	0.451	0.460	0.436	0.459	0.444	0.462	0.467	0.464	0.45
Winogrande	0.522	0.501	0.510	0.503	0.508	**0.525**	0.520	0.515	0.512	0.519	0.508	0.500	0.512	0.505	0.510	0.50
BLiMP	0.810	0.727	0.738	0.746	0.748	0.749	0.802	0.814	0.809	0.779	0.821	0.817	0.822	0.738	**0.826**	0.50
MultiBLiMP	0.971	0.919	0.910	0.856	0.925	0.927	0.975	**0.987**	0.982	0.969	**0.987**	0.978	0.983	0.869	0.982	0.50
EWoK	0.566	0.518	0.518	0.537	0.522	0.527	0.556	0.566	0.574	0.544	0.575	**0.583**	0.557	0.543	0.563	0.50
PIQA	0.661	0.596	0.597	0.613	0.608	0.611	0.641	0.653	0.655	0.640	0.655	0.665	**0.666**	0.609	0.661	0.50
HellaSwag	0.327	0.293	0.291	0.295	0.293	0.300	0.320	0.325	0.321	0.311	0.322	0.324	**0.328**	0.298	**0.327**	0.25
PROST	0.207	0.240	0.225	0.239	0.221	**0.252**	0.212	0.234	0.232	0.218	0.218	0.218	0.209	0.249	0.208	0.25
MMLU	0.234	0.232	0.230	0.233	0.233	0.237	0.235	**0.248**	0.237	0.233	0.237	0.232	0.238	0.231	0.232	0.25
BBH	0.000	0.000	0.000	0.002	0.000	0.002	0.002	0.000	**0.005**	0.000	0.000	0.002	0.001	0.000	0.000	0.33
XQuAD	0.000	0.000	0.000	0.010	0.000	0.000	0.001	0.000	0.000	0.000	**0.019**	0.002	0.000	0.000	0.000	0.00
Belebele	0.334	0.322	0.314	0.313	0.324	0.283	0.329	**0.364**	0.330	0.343	0.334	0.344	0.334	0.328	0.337	0.25
COMPS	0.599	0.519	0.519	0.551	0.519	0.559	0.562	0.599	0.594	0.549	0.596	0.596	**0.602**	0.553	0.601	0.50
ANALOGICAL	0.619	0.445	0.492	0.410	0.511	0.364	0.629	0.632	0.590	0.447	0.644	0.590	0.587	0.414	**0.652**	0.25
Avg.	0.452	0.412	0.414	0.412	0.420	0.414	0.446	**0.456**	0.450	0.428	0.455	0.450	0.450	0.415	0.454	0.36

D.3 VISTA

The results for each of the VISTA models are presented by language: Amharic (Table 19), English 130M (Table 20), English 350M (Table 21), Vietnamese (Table 22), Thai (Table 23), and Urdu (Table 24).

Table 19: Evaluation scores for Amharic VISTA models on Amharic tasks.

Tokenizer Type	Vocab Size	AfriMMLU	Belebele	Global MMLU	MultiBLiMP	Global PIQA	Avg.
BPE	8192	0.222	0.253	0.236	0.946	**0.520**	0.402
	16384	0.232	0.254	0.249	0.955	0.470	0.402
	32768	0.230	0.268	0.251	0.893	0.500	0.399
	49152	0.240	0.253	0.253	**1.000**	0.450	0.408
	65536	0.228	0.260	**0.260**	0.973	0.480	**0.410**
	81920	**0.248**	0.257	0.248	0.973	0.470	0.407
	98304	0.222	0.262	0.255	0.964	0.450	0.401
	114688	0.240	**0.271**	0.239	0.964	0.420	0.396
Unigram	8192	0.240	0.257	0.230	0.982	0.450	0.398
	16384	0.232	0.260	0.232	0.920	0.450	0.387
	32768	0.224	0.258	0.243	0.955	0.450	0.396
	49152	0.232	0.262	0.239	0.982	0.430	0.398
	65536	0.228	0.249	0.230	0.973	0.430	0.390
	81920	0.226	0.260	0.230	0.991	0.380	0.386
	98304	0.226	0.257	0.229	0.982	0.410	0.389
	114688	0.228	0.253	0.230	0.938	0.400	0.380
Chance		0.25	0.25	0.25	0.50	0.50	0.33

Table 20: Evaluation scores for English VISTA models on English tasks. ARC Challenge and ARC Easy are abbreviated to ARC-C and ARC-E.

Tokenizer Type	Vocab Size	ARC-C	ARC-E	Belebele	BLiMP	COPA	Global MMLU	Hella Swag	Multi BLiMP	Winogrande	XWinograd	Global PIQA	Super GLUE	Avg.
BPE	8192	0.191	0.277	**0.273**	0.586	0.580	**0.232**	**0.266**	0.818	0.488	**0.523**	0.520	**0.463**	0.419
	16384	0.182	0.269	0.252	0.600	0.580	0.230	0.259	0.783	0.479	0.514	0.450	0.434	0.405
	32768	0.188	0.280	0.258	0.613	0.520	0.229	0.255	0.832	0.511	0.505	**0.550**	0.439	0.416
	49152	0.181	0.270	0.248	0.654	0.550	0.229	0.255	0.810	0.503	0.508	0.480	0.435	0.412
	65536	0.177	0.275	0.251	0.640	0.540	0.230	0.255	0.838	0.505	0.497	0.470	0.447	0.412
	81920	0.178	0.277	0.257	0.645	0.510	0.230	0.258	**0.847**	0.503	0.511	0.510	0.434	0.415
	98304	0.171	0.279	0.260	**0.661**	0.510	0.231	0.260	0.842	0.507	0.512	0.510	0.438	0.416
	114688	0.176	0.281	0.266	0.658	0.540	0.230	0.258	0.836	0.494	0.509	0.500	0.442	0.417
Unigram	8192	0.200	0.270	0.256	0.587	0.580	0.231	0.265	0.769	0.507	0.519	0.490	0.452	0.412
	16384	**0.201**	**0.287**	0.252	0.602	0.590	0.230	0.257	0.806	0.486	0.513	0.490	0.446	0.415
	32768	0.187	0.268	0.253	0.618	0.480	0.230	0.259	0.804	0.486	0.493	0.500	0.403	0.401
	49152	0.186	0.274	0.254	0.640	**0.630**	0.229	0.258	0.797	0.497	0.517	0.500	0.441	**0.420**
	65536	0.179	0.271	0.258	0.626	0.570	0.230	0.257	0.805	0.502	0.505	0.510	0.444	0.414
	81920	0.186	0.272	0.258	0.602	0.560	0.229	0.257	0.812	0.502	0.503	0.510	0.453	0.413
	98304	0.189	0.266	0.254	0.639	0.600	0.229	0.258	0.786	0.487	0.521	0.520	0.424	0.416
	114688	0.199	0.274	0.259	0.592	0.600	0.230	0.259	0.796	**0.537**	0.513	0.500	0.433	0.417
Chance		0.25	0.25	0.25	0.50	0.50	0.25	0.25	0.50	0.50	0.50	0.50	0.48	0.38

Table 21: Evaluation scores for English VISTA 350M models on English tasks.

Tokenizer Type	Vocab Size	GLUE	Wino-grande	BLiMP	Multi BLiMP	EWoK	PIQA	Hella Swag	PROST	MMLU	BBH	XQuAD	Bele-bele	COMPS	ANA-LOGICAL	Avg.
BPE	8192	**0.469**	0.499	0.593	**0.861**	0.525	0.612	0.289	0.269	**0.233**	0.059	0.000	0.288	0.540	0.398	0.402
	32768	0.456	0.501	0.624	0.825	0.530	0.595	0.294	0.217	0.231	0.144	0.005	0.301	0.526	0.368	0.401
	65536	0.466	**0.516**	0.579	0.792	0.535	0.606	0.292	0.252	0.229	0.163	**0.008**	**0.321**	0.551	0.377	0.406
	98304	**0.469**	0.497	0.629	0.855	0.540	**0.630**	**0.301**	**0.272**	0.230	**0.181**	0.001	0.293	**0.575**	**0.448**	**0.423**
Unigram	8192	0.413	0.490	0.582	0.852	0.512	0.582	0.283	0.244	0.230	0.090	0.000	0.298	0.512	0.362	0.389
	32768	0.444	0.511	0.595	0.816	0.524	0.611	0.294	0.260	**0.232**	0.134	0.000	0.278	0.542	0.371	0.401
	65536	0.462	0.509	0.584	0.810	**0.543**	0.616	0.291	0.260	0.230	0.134	**0.008**	0.290	0.544	0.445	0.409
	98304	0.396	0.506	**0.633**	0.848	**0.546**	0.604	0.299	0.267	0.228	0.127	0.007	0.294	0.569	0.436	0.411
Chance		0.45	0.50	0.50	0.50	0.50	0.50	0.25	0.25	0.25	0.33	0.00	0.25	0.50	0.25	0.36

Table 22: Evaluation scores for Vietnamese VISTA models on Vietnamese tasks.

Tokenizer Type	Vocab Size	ARC	Belebele	Global MMLU	HellaSwag	INCLUDE	ViQuAD	VMLU	XCOPA	XNLI	Global PIQA	Avg.
bpe	8192	0.209	**0.256**	0.245	**0.266**	0.255	0.006	0.249	0.544	0.343	0.520	0.285
	16384	0.206	0.228	0.248	0.260	0.253	0.016	0.270	0.540	0.341	0.560	0.288
	32768	0.206	0.239	0.246	0.262	0.260	0.031	0.277	0.532	0.356	0.580	0.294
	49152	**0.220**	0.239	0.249	0.259	0.253	0.021	0.263	0.530	0.337	0.600	0.293
	65536	0.214	0.246	0.241	0.259	0.273	0.001	**0.290**	0.534	0.341	0.550	0.290
	81920	0.212	0.232	0.242	0.257	0.267	0.056	0.267	0.506	0.330	0.590	0.291
	98304	0.215	0.241	0.238	0.258	0.275	0.038	0.250	0.526	0.334	0.570	0.289
	114688	0.213	0.247	0.240	0.259	**0.278**	0.062	0.282	0.532	0.337	**0.610**	**0.300**
unigram	8192	0.199	0.240	0.242	0.265	0.276	0.024	0.262	0.532	0.333	0.530	0.286
	16384	0.203	0.244	0.249	0.262	0.236	**0.195**	0.273	0.540	0.353	0.490	0.299
	32768	0.209	0.244	0.250	0.259	0.249	0.037	0.253	0.540	**0.367**	0.540	0.291
	49152	0.209	0.243	0.252	0.260	0.238	0.040	0.265	**0.548**	0.361	0.560	0.293
	65536	0.212	0.242	0.243	0.258	0.233	0.035	0.246	0.536	0.340	0.540	0.284
	81920	0.205	0.242	**0.254**	0.262	0.227	0.047	0.270	**0.548**	0.345	0.580	0.294
	98304	0.205	0.249	0.252	0.262	0.225	0.031	0.265	0.540	0.334	0.530	0.286
	114688	0.207	0.238	0.254	0.259	0.231	0.084	0.273	0.538	0.341	0.530	0.292
Chance	–	0.25	0.25	0.25	0.25	0.25	0.00	0.25	0.50	0.33	0.50	0.28

Table 23: Evaluation scores for Thai VISTA models on Thai tasks.

Tokenizer Type	Vocab Size	Belebele	Winograd	XCOPA	MMLU	Global PIQA	Avg.
BPE	8192	0.259	0.470	0.536	0.082	0.540	0.377
	16384	0.261	**0.540**	0.556	0.083	0.500	0.388
	32768	0.251	0.530	0.544	0.080	0.540	0.389
	49152	0.246	**0.540**	0.540	0.084	0.530	0.388
	65536	0.254	0.516	0.562	0.082	0.510	0.385
	81920	0.256	0.509	0.550	0.083	0.510	0.381
	98304	0.253	0.498	0.550	0.082	0.500	0.377
	114688	0.247	0.519	0.540	0.084	0.490	0.376
Unigram	8192	**0.278**	0.474	0.564	0.081	0.490	0.377
	16384	0.263	0.495	0.528	0.085	0.510	0.376
	32768	0.259	0.526	0.544	0.083	0.510	0.384
	49152	0.247	0.519	0.554	**0.085**	**0.550**	**0.391**
	65536	0.259	0.456	0.564	0.084	0.500	0.373
	81920	0.247	0.533	0.562	0.081	0.530	0.391
	98304	0.252	0.512	**0.576**	0.083	0.490	0.383
	114688	0.248	0.512	0.570	0.082	0.520	0.387
Chance		0.25	0.50	0.50	0.25	0.50	0.40

Table 24: Evaluation scores for Urdu VISTA models on Urdu tasks.

Tokenizer Type	Vocab Size	Belebele	MMLU ProX	Global PIQA	Avg.
BPE	8192	0.249	0.085	0.590	0.308
	16384	0.259	0.083	**0.620**	**0.321**
	32768	0.254	0.086	**0.620**	0.320
	49152	0.254	0.086	0.610	0.317
	65536	**0.262**	0.086	0.610	0.319
	81920	0.244	0.087	0.580	0.304
	98304	0.253	0.087	**0.620**	0.320
	114688	0.253	0.086	0.600	0.313
Unigram	8192	0.242	**0.088**	0.560	0.297
	16384	0.242	0.088	0.550	0.293
	32768	0.248	0.087	0.560	0.298
	49152	0.254	0.087	0.530	0.290
	65536	0.249	0.088	0.550	0.296
	81920	0.252	0.087	0.530	0.290
	98304	0.247	0.086	0.530	0.287
	114688	0.247	**0.088**	0.560	0.298
Chance		0.25	0.10	0.50	0.28

D.3.1 Correlation Between Vocab Size and Performance

Figure 7 plots the relationship between vocabulary size (log-scaled; x-axis) and aggregate performance (mean for each subset, and macro average across language; y-axis) for the VISTA models. For each facet, Spearman's ρ and p-value are reported.

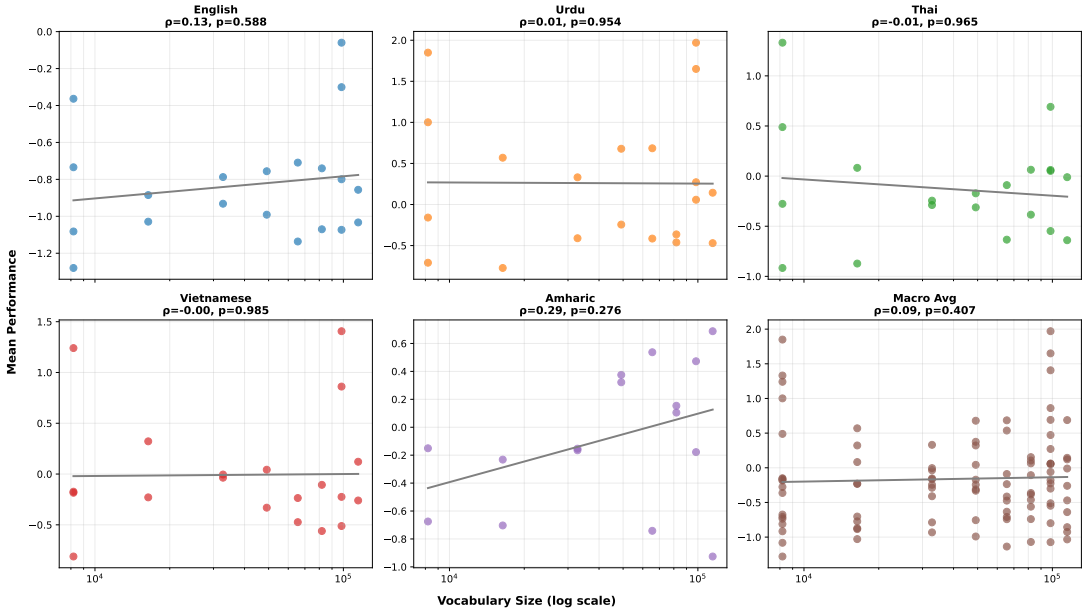


Figure 7: Correlation between vocabulary size (x-axis) and aggregated (z-scored) model performance by language and macro-averaged over language (bottom right facet). Each facet also reports correlation coefficient and p-value.

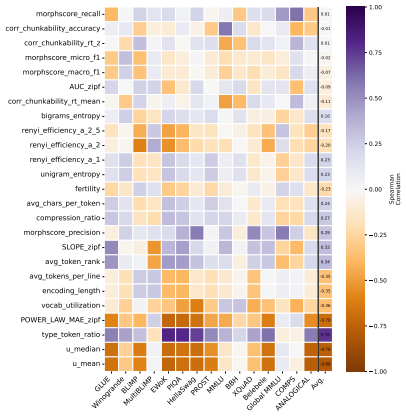
E Full Metric-Performance Correlation Results

Here we show the correlation results for all of the model suites and all of the languages. Figure 8 shows the correlations for the TokSuite models. Figure 9 shows the correlations for the TIMTC models. Figure 10 shows the correlations for the VISTA models.

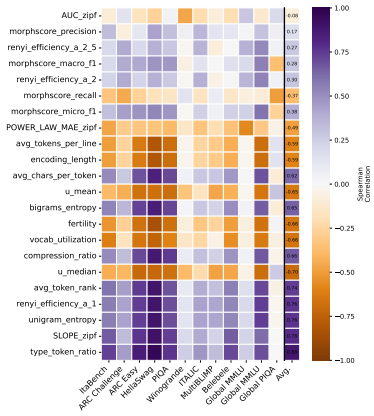
In Table 25, we report the by-metric correlation reports over all tasks (left) and tasks for which the 130M parameter models scored above chance (right; see Section F).

Table 25: Correlation between tokenizer metrics and model performance, with and without task exclusions (for tasks where performance was near chance). Significance markers (*) indicate $p_{\text{corrected}} < 0.05$.

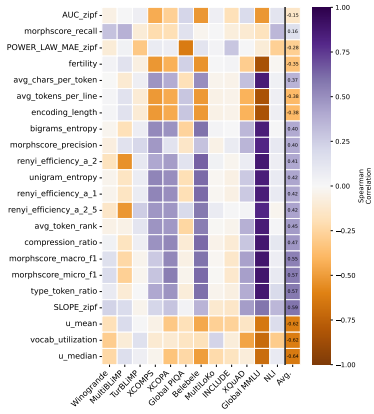
	All Tasks				With Exclusions			
Metric	r	p	p_{corr}	sig	r	p	p_{corr}	sig
AUC_zipf	0.03	0.17	0.26		−0.10	0.23	0.44	
avg_chars_per_token	−0.00	0.83	0.98		−0.24	0.00	0.02	*
avg_token_rank	0.03	0.16	0.26		−0.03	0.70	0.94	
avg_tokens_per_line	0.02	0.36	0.45		0.05	0.53	0.84	
bigrams_entropy	0.00	0.89	0.98		−0.10	0.23	0.44	
compression_ratio	−0.05	0.02	0.04	*				
corr_chunkability_accuracy	−0.08	0.00	0.00	*	−0.25	0.00	0.02	*
corr_chunkability_rt_mean	0.02	0.23	0.31		0.14	0.09	0.25	
corr_chunkability_rt_z	0.03	0.07	0.15		0.15	0.06	0.17	
encoding_length	0.03	0.12	0.21					
fertility	0.02	0.23	0.31		0.04	0.61	0.92	
final_tokens_count	0.18	0.00	0.00	*	0.17	0.03	0.10	
gini_coefficient	0.19	0.00	0.00	*	0.19	0.02	0.08	
leading_space_tokens_count	0.27	0.00	0.00	*	0.27	0.00	0.02	*
leading_space_tokens_share	0.22	0.00	0.00	*	0.19	0.02	0.08	
morphscore_macro_f1	−0.09	0.00	0.00	*	0.00	0.96	1.00	
morphscore_micro_f1	−0.09	0.00	0.00	*	0.00	1.00	1.00	
morphscore_precision	−0.07	0.00	0.00	*	−0.13	0.14	0.33	
morphscore_recall	−0.12	0.00	0.00	*	−0.09	0.28	0.49	
POWER_LAW_MAE_zipf	−0.03	0.09	0.18		−0.00	0.99	1.00	
renyi_efficiency_a_1	0.02	0.24	0.31		−0.03	0.75	0.94	
renyi_efficiency_a_2	0.00	1.00	1.00		0.00	0.98	1.00	
renyi_efficiency_a_2_5	−0.00	0.88	0.98		−0.01	0.95	1.00	
SLOPE_zipf	−0.04	0.06	0.12		−0.18	0.02	0.09	
type_token_ratio	0.03	0.15	0.25		0.06	0.49	0.82	
u_mean	−0.04	0.02	0.05		0.04	0.66	0.94	
u_mean_log_weighted	−0.00	0.97	1.00		0.12	0.14	0.33	
u_mean_weighted	0.00	0.96	1.00		0.11	0.19	0.41	
u_median	−0.05	0.01	0.04	*	0.00	0.96	1.00	
unigram_entropy	0.02	0.24	0.31		−0.03	0.75	0.94	
used_in_merges	0.24	0.00	0.00	*	0.25	0.00	0.02	*
vocab_utilization	−0.14	0.00	0.00	*	−0.20	0.01	0.07	



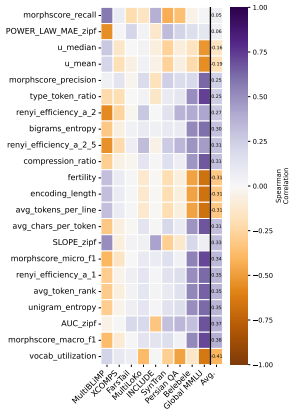
(a) English



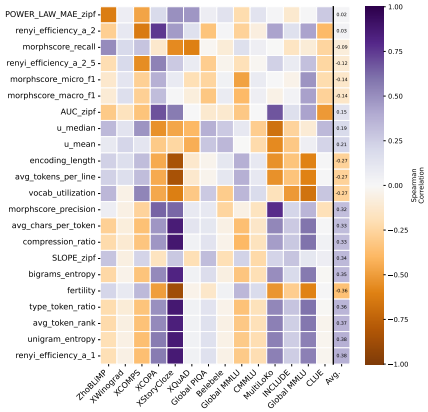
(b) Italian



(c) Turkish



(d) Farsi



(e) Chinese

Figure 8: Correlations between intrinsic metric and model performance for TokSuite models for each of the five languages covered. Each column represents an individual task, with the final column representing aggregated performance. The color of each cell corresponds to direction of correlation and the shade corresponds to the strength.

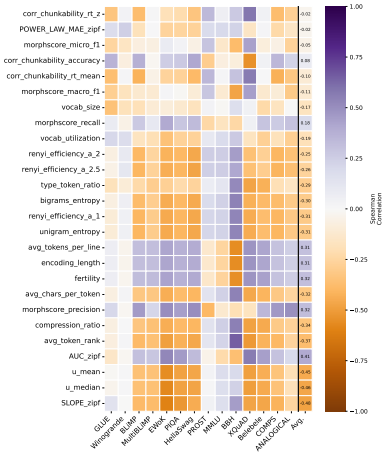
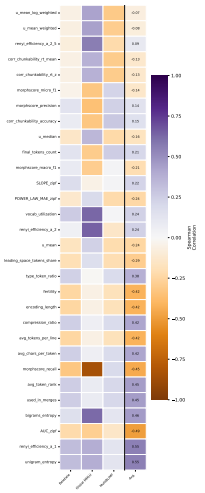
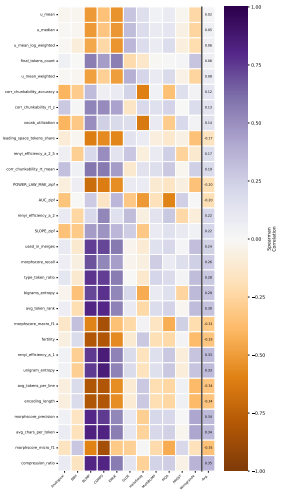


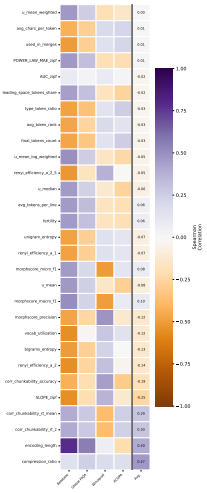
Figure 9: Correlations between intrinsic metric and model performance for the TIMTC models for English. Each column represents an individual task, with the final column representing aggregated performance. The color of each cell corresponds to direction of correlation and the shade corresponds to the strength.



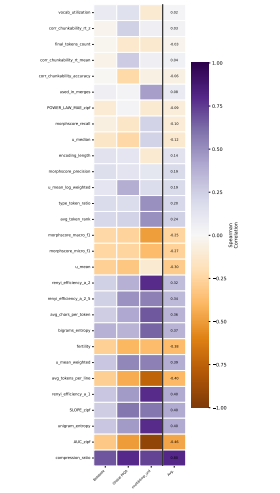
(a) Amharic



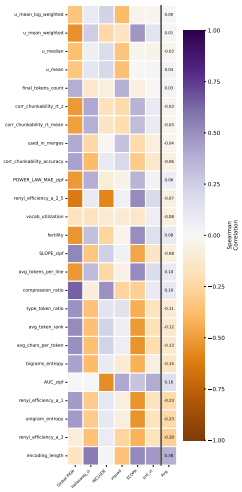
(b) English



(c) Thai



(d) Urdu



(e) Vietnamese

Figure 10: Correlations between intrinsic metric and model performance for VISTA models for each of the five languages. Each column represents an individual task, with the final column representing aggregated performance. The color of each cell corresponds to direction of correlation and the shade corresponds to the strength.

E.1 Correlation with Orthographic Awareness Tasks

Results in Figure 11 show performance on the orthographic tasks. The ByT5 tokenizer model performs best on a subset of the orthographic tasks (namely, CUTE and EXECUTE), though, in aggregate, it does not outperform other models. As all the orthographic tasks are similar and are composed of heterogeneous subtasks, it is not clear why the byte-level model outperforms the other models on some tasks but not others.

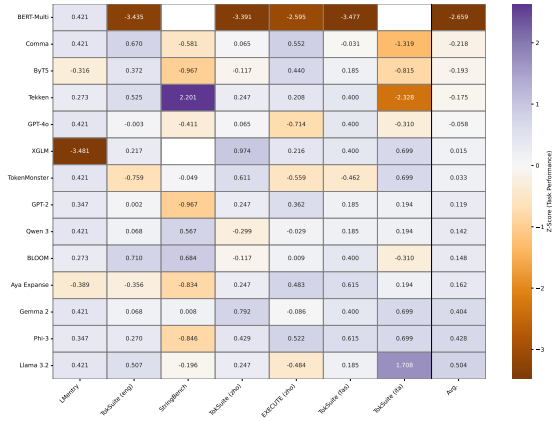


Figure 11: Performance for each of the TokSuite models for the English orthographic awareness tasks. Color corresponds to the z-scored performance, where darker green indicates better performance.

Table 26 presents the correlations between performance on each of the orthographic tasks and downstream tasks in the corresponding language.

Task Group	Language	ρ	p-value
LMentry	English	-0.179	0.5408
TokSuite	English	0.341	0.0292
StringBench	English	0.163	0.4806
TokSuite	Chinese	0.114	0.6990
EXECUTE	Chinese	0.267	0.0663
TokSuite	Farsi	0.433	0.1224
TokSuite	Italian	-0.070	0.8199

Table 26: Correlation between orthographic tasks and downstream tasks.

E.2 Multiple Predictor Correlations

Figure 12 shows the correlations between each of the intrinsic tokenizer metrics. Within each type of tokenizer evaluation (indicated with black boxes), many metrics are strongly correlated with each other. Outside of those types, there are several other strong correlations. For example both unigram and bigram entropy are strongly correlated with compression metrics, and MorphScore is somewhat correlated with worse compression, as predicted. Many potential relationships emerge in this analysis. So, in the next section, we test whether combining any of these metrics can better predict downstream performance.

We test whether a combination of two intrinsic tokenizer metrics better predicts model performance. Table 27 shows the variance explained (adjusted R^2) by linear regressions with individual metrics (`adj_r2_m1` and `adj_r2_m2`) as predictors and both metrics as predictors (`adj_r2_joint`) for the top ten combinations. `r2_gain` shows the increased explained variance with the two predictors together. The p-value was corrected for multiple comparisons.

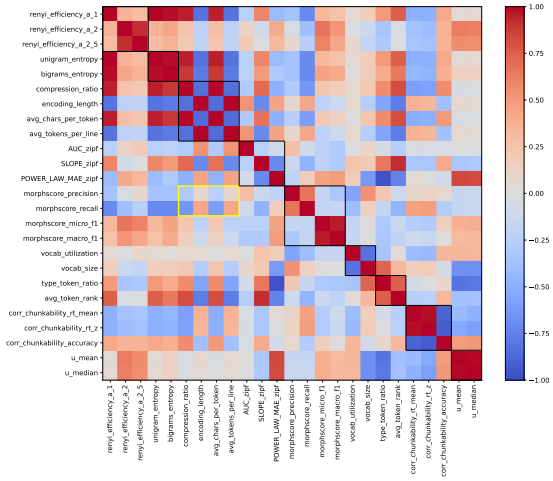


Figure 12: The boxes highlight comparisons between metrics that index similar things. Yellow highlights correlation between MorphScore and compression metrics.

F Exclusion Analysis

In this section, we compare the results with and without the exclusion of tasks where most models did not perform above chance. We excluded tasks for which there was not more than one model from each of the model families that achieved above-chance performance. Table 25 compares the by-metric correlations with task performance with and without the exclusions. Table 28 shows the results for the correlations with two predictors (cf. Table 27).

Table 27: Variance explained by combinations of two metrics, shown for the ten best combinations.

	Metric 1	Metric 2	adj_r2_m1	adj_r2_m2	adj_r2_joint	r2_gain	p-value
1	leading_space_tokens_count	u_mean_log_weighted	0.158	−0.000	0.170	0.012	< 0.001
2	final_tokens_count	used_in_merges	0.053	0.110	0.168	0.058	< 0.001
3	leading_space_tokens_count	u_mean	0.158	0.002	0.167	0.009	< 0.001
4	leading_space_tokens_count	u_median	0.158	0.003	0.165	0.007	< 0.001
5	leading_space_tokens_count	u_mean_weighted	0.158	0.001	0.165	0.007	< 0.001
6	leading_space_tokens_count	AUC_zipf	0.124	0.011	0.157	0.033	< 0.001
7	used_in_merges	u_mean_log_weighted	0.147	−0.000	0.154	0.007	< 0.001
8	used_in_merges	u_mean_weighted	0.147	0.001	0.152	0.005	< 0.001
9	used_in_merges	u_mean	0.147	0.002	0.151	0.004	< 0.001
10	used_in_merges	u_median	0.147	0.003	0.149	0.002	< 0.001

Table 28: Variance explained by combinations of two metrics, shown for the ten best combinations. These results exclude any of the tasks where the small models performed near chance.

	Metric 1	Metric 2	adj_r2_m1	adj_r2_m2	adj_r2_joint	r2_gain	p-value
1	leading_space_tokens_count	u_mean_log_weighted	0.074	−0.000	0.085	0.012	< 0.001
2	leading_space_tokens_count	u_mean_weighted	0.074	0.000	0.084	0.010	< 0.001
3	leading_space_tokens_count	u_mean	0.074	0.002	0.083	0.009	< 0.001
4	leading_space_tokens_count	u_median	0.074	0.003	0.082	0.008	< 0.001
5	leading_space_tokens_count	compression_ratio	0.072	0.000	0.073	0.000	< 0.001
6	leading_space_tokens_count	encoding_length	0.072	0.000	0.072	0.000	< 0.001
7	leading_space_tokens_count	morphscore_macro_f1	0.065	0.004	0.070	0.004	< 0.001
8	leading_space_tokens_count	morphscore_micro_f1	0.065	0.004	0.069	0.004	< 0.001
9	used_in_merges	u_mean_weighted	0.062	0.000	0.068	0.007	< 0.001
10	used_in_merges	u_mean_log_weighted	0.062	−0.000	0.068	0.006	< 0.001

G AI Use

The authors used AI to aid in code generation for figure and table creation and to edit the text of this paper. The authors take full responsibility for the content of the paper and any associated artifacts.