DISENTANGLING TOKENIZER CONFOUNDS IN BILINGUAL TRANSFER ESTIMATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Training a multilingual language model requires choosing how much data to use from each language, and a language trained alongside a target language can speed up or slow down its learning. Longpre et al. (2026) quantify this with the Bilingual Transfer Score (BTS): the number of tokens a 50/50 bilingual run needs, relative to a monolingual baseline, to reach the same target-language loss. Using a large set of empirical language modeling runs, they provide pairwise BTS estimates across many language pairs as a reusable resource for selecting data mixtures. Notably, all scores were estimated under a single multilingual tokenizer. It has been observed that intrinsic tokenizer metrics differ widely across languages (Petrov et al., 2023; Ahia et al., 2023), which raises the question: to what extent does a pair's score reflect the language pair vs. the tokenizer used to estimate it? Holding the model and training data fixed, we recompute BTS under tokenizers that differ in which of the two languages they are trained on, and analyze how the score changes with tokenizer training data composition, per-language fragmentation, and script. We confirm that tokenizer training data composition and language fragmentation are non-negligible factors systematically associated with the estimates. We then define the Tokenizer-adjusted BTS, which explicitly controls for tokenizer-induced variation. We re-estimate BTS for several language pairs using this method and show that it differs significantly from the original estimation method.

1 INTRODUCTION

Scaling laws have become an integral part of language model training, allowing developers to allocate model size and data without searching configurations at full scale (Kaplan et al., 2020; Hoffmann et al., 2022). For multilingual models, however, determining the ideal training settings is more complex than a simple function of the number of model parameters, total token counts and compute budget. Complex learning dynamics arise when training with multilingual data mixtures: some language pairs exhibit cross-lingual transfer while others are at odds with each other, showing cross-lingual interference (Alastruey et al., 2025). Longpre et al. (ATLAS; 2026) take an important step in explicitly accounting for these dynamics in scaling laws. From their empirical study across 774 training runs spanning 400+ training languages and 48 evaluation languages, they estimate **Bilingual Transfer Scores (BTS)** between language pairs: the number of tokens a 50/50 bilingual run requires, relative to a monolingual baseline, to reach the same target-language loss. They use these scores in the parameter estimation of their proposed Adaptive Transfer Scaling Law—a multilingual variant of standard scaling laws that accounts for cross-lingual transfer and interference dynamics.

Language-pair BTS estimates are a main contribution of Longpre et al. (2026), offered to the community as a reusable resource for optimizing data mixture choices for multilingual language model training. Their reusability, though, depends on the entries not varying strongly with model design choices that the Adaptive Transfer Scaling Law leaves unspecified. We hypothesize that tokenizer choice violates this assumption. The tokenizer directly determines per-language token counts and vocabulary sharing; the former is baked into the BTS equation and the ATLAS authors hypothesize that the latter is a large factor in what leads to high scores between pairs. These tokenizer attributes can vary wildly across both languages and tokenizers (Rust et al., 2021; Petrov et al., 2023; Ali et al., 2024), suggesting that BTS estimates may be sensitive to them. The one comparison where the vocabulary is varied in Longpre et al. (2026) provides initial evidence in support of this; their Figure 1 shows that the per-language scaling frontier—the relationship that BTS is built—shifts

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

notably when changing between mono and multilingual tokenizers. We therefore hypothesize that tokenizer choice needs to be accounted for to derive generalizable BTS scores.

In this work, we first investigate empirically the degree to which BTS scores are affected by tokenizer choice. To this end, we estimate the score under tokenizers that differ in which of the two languages they are trained on, the source, the target, both, or neither, with model and data held fixed. We perform a detailed analysis of how the score changes with tokenizer training data composition, per-language fragmentation, and script. We then propose a method to create tokenizer-adjusted BTS estimates, which incorporates tokenizer properties into BTS modeling, therefore explicitly separating pair-level transfer from tokenizer artifacts. As a result, we get BTS estimates that better hold across tokenizers, providing a transfer estimate that is not tied to a single tokenizer.

2 Related Work

2.1 Multilingual LLMs

Multilingual Tokenization. Tokenization is crucial for multilingual LLMs, as the tokenizer serves as the bridge between human language and the model's latent space and thus determines how each language is segmented and represented (Velayuthan & Sarveswaran, 2025); moreover, the representational biases introduced at this stage have been shown to correlate with, and even predict, downstream performance across languages (Kanjirangat et al., 2025). To improve the efficiency and performance of language models, previous work has explored various tokenization algorithms (Hu et al., 2025; Liu et al., 2025; Erdogan et al., 2026), several focusing specifically on multilingual capabilities and cross-lingual fairness (Limisiewicz et al., 2024; Ahia et al., 2024; Foroutan et al., 2025; Abagyan et al., 2025).

Cross-Lingual Transfer. A central question in multilingual modeling is how knowledge is shared across languages. At the macro level, recent scaling-law studies quantify mutual-benefit scores between language pairs and the curse of multilinguality (Longpre et al., 2026). At the tokenization level, several works reshape the vocabulary to enable transfer: parallel tokenizers align monolingual vocabularies so that semantically equivalent words share indices (Kautsar & Koto, 2025), and analyses of vocabulary overlap show that shared—especially semantically similar—tokens improve cross-lingual transfer (Kallini et al., 2025). A complementary line adapts a pre-trained model to new languages via cross-lingual vocabulary transfer, initializing target embeddings from related source embeddings using translation resources (Remy et al., 2024) or bilingual dictionaries (Sakajo et al., 2025). More broadly, tokenizer design and specialization during fine-tuning also affect downstream and adaptation performance (Dagan et al., 2024). While these lines study tokenization and cross-lingual transfer separately, none treats the tokenizer as a confound in scaling-law–based transfer estimates themselves—the gap this work addresses.

2.2 Tokenizer Intrinsic Metrics

Comparing tokenizers by their downstream effect requires training models, which is prohibitively expensive; this has motivated *intrinsic* metrics, computable from the tokenizer or tokenized text alone, to predict tokenizer quality cheaply and offer an interpretable account of what makes a segmentation good (Zouhar et al., 2023). While intrinsic metrics do not always reliably predict downstream performance, they hold great potential for cheap and interpretable tokenizer evaluation.[1]

Existing metrics (see Appendix A) can be categorized into different families. The most common are compression-based, such as fertility and compression rate; however, the assumption that fewer tokens yield better models is contested, with some work finding minimal segmentation unhelpful and others reporting that compression does correlate with downstream success (Schmidt et al., 2024; Goldman et al., 2024). A second, information-theoretic family treats tokenization as an efficient channel usage, proposing the Rényi efficiency of the unigram distribution as a predictor of performance (Zouhar et al., 2023). More recently, token-bigram statistics are also considered training-free proxies for language-modeling difficulty (Poelman et al., 2025). However, these metrics also admit counterexamples (Cognetta et al., 2024). A third, linguistically motivated family asks whether token boundaries respect meaningful units, measuring alignment with morphemes, cognitive plausibility, and the effect of token

[1] Open-source tools, such as TokEval (Meister, 2025) are widely-available for this.

granularity on surprisal-based models of human reading (Arnett & Bergen, 2025; Arnett et al., 2026; Beinborn & Pinter, 2023; Oh & Schuler, 2025). Finally, multilingual work focuses on cross-lingual equity through metrics such as vocabulary allocation and overlap, single-token retention rate (STRR), and token alignability across scripts (Limisiewicz et al., 2023; Tafseer Nayeem et al., 2025; Hämmerl et al., 2025).

3 Bilingual Transfer Score

3.1 Original Definition of BTS in ATLAS

The bilingual transfer score (BTS) is a concept introduced by Longpre et al. (2026). In words, the BTS is meant to quantify how much a source language s accelerates learning of a target language t compared with monolingual training, allowing developers to estimate the amount of training data tokens needed to reach a specified performance level for a multilingual language model.

Concretely, given a target-language training budget d_{mono}, let $L_t(d_{mono})$ be the target-language validation Bits-Per-Byte (BPB) reached after training a monolingual with d_{mono} target tokens. We then ask how many total tokens we'd need in order to reach this same validation Bits-Per-Byte on t when training a bilingual model, which is denoted by $\sigma_{bi}\big(L_t(d_{mono})\big)$. The $\text{BTS}^T_{s\to t}$ measures the deviation of this quantity from the number of tokens needed in the monolingual baseline:[2]

$$\text{BTS}_{s\to t} = -\frac{\sigma_{bi}\big(L_t(d_{mono})\big) - 2d_{mono}}{d_{mono}}. \tag{1}$$

Intuitively, a bilingual run that reaches the monolingual BPB with fewer than $2d_{mono}$ total tokens ($\sigma_{bi} < 2d_{mono}$) benefits from the source language and yields a positive score. The normalization by d_{mono} makes the score comparable across language pairs and centers the no-transfer case at 0: in that case, the extra d_{mono} source-language tokens provide no benefits, so the bilingual one needs exactly $2d_{mono}$ total tokens to match the monolingual target loss.

The original definition of BTS (eq. (1)) suggests that BTS should be an intrinsic property of the language pair (s, t), and that scores would be generally applicable for capturing cross-lingual transfer in language models, independent of model hyperparameter settings and design choices.[3] We hypothesize that the model's tokenizer is a design choice that breaks BTS score generalizability. Concretely, the tokenizer impacts estimated BTS in two clear ways. First, BTS is a ratio of training-token counts, where tokens come from two different languages; different languages are tokenized at different rates and the ratio does not normalize for this difference.[4] The vocabulary-insensitive loss that Longpre et al. (2026) use as the target (Tao et al., 2024) does not remove this dependence: it makes the loss value comparable across vocabularies, but the token budgets and mixing ratios implicitly remain in the units of the tokenizer used. Second, the overlap in subword units between two languages, which Longpre et al. (2026) identify as the likely mechanism behind the higher scores of same-script pairs, is set by the vocabulary; two tokenizers over the same languages can share more or fewer subwords. Together, these point to the tokenizer as an axis with large potential to affect BTS scores, but it is a design choice held fixed in the estimates of Longpre et al. (2026).

To explicitly account for this tokenizer dependence, we first replace the token budget d_{mono} with the **byte budget** b_{mono} to ensure the same amount of training data as a fixed reference point across tokenizers with different compression rates. Then we condition the training-driven variables L_t and σ on the exact tokenizer used T by rewriting them as $L^T_t(b_{mono})$ and $\sigma^T_{bi}\big(L^T_t(b_{mono})\big)$. The final

[2] Because only half of its tokens are in the target language, without any cross-lingual transfer, the bilingual model would need $2d_{mono}$ tokens to expose the model to d_{mono} target tokens, hence the multiplicative factor in the denominator.

[3] Most scaling laws are defined in a similar manner, using only a minimal set of variables that are typically independent of language model design choices (Kaplan et al., 2020; Hoffmann et al., 2022; He et al., 2024).

[4] For parallel text, the number of tokens differs across languages by up to a factor of 15 under a fixed tokenizer (Petrov et al., 2023), and these rates are not stable across tokenizers (Rust et al., 2021). For instance, the Qwen 3 tokenizer, whose vocabulary is ∼18% larger than Llama 3.3's (Grattafiori et al., 2024; Team, 2025), encodes the FLORES-200 Ukrainian subset into 54.7% more tokens than Llama 3.3 does, though both are multilingual tokenizers. Differences can be larger for language-specialized tokenizers (Ovcharov, 2026).

transfer score is thus also conditioned on T:

$$\text{BTS}^T_{s\to t} = -\frac{\sigma^T_{bi}\big(L^T_t(b_{mono})\big) - 2b_{mono}}{b_{mono}}. \tag{2}$$

3.2 Tokenizer-adjusted Bilingual Transfer Score (TBTS)

Results in Longpre et al. (2026) indicate that the choice of tokenizer is not neutral. Training with a multilingual rather than a per-language vocabulary changes the relationship between loss and compute (their Figure 1). Our results in Sections 5–5.1 corroborate this: the same language pair receives systematically different $\text{BTS}^T_{s\to t}$ depending on the tokenizer, indicating that a single $\text{BTS}^T_{s\to t}$ is tied to one tokenizer. To obtain an estimate that generalizes across tokenizers, we model $\text{BTS}^T_{s\to t}$ as a function of tokenizer intrinsic properties to account for tokenizer influence in transfer.

Regression Model. Let $\text{BTS}^T_{s\to t}$ be the score for language pair s, t measured under tokenizer T. Our goal is to decompose each tokenizer-specific $\text{BTS}^T_{s\to t}$ into (1) a transfer performance inherent to each pair, and (2) offset introduced by the tokenizer. Let $u_{s\to t}$ refer to the intrinsic transfer level of the pair s, t—our operationalization of (1). Further, let $x(T, s)$, $x(T, t)$ and $p(T, s, t)$ be intrinsic metrics for tokenizer T; the first two are computed on the source and target languages, individually, while the last is a pairwise metric. Together, these serve as the variables that enable us to capture (2). In practice, all three tokenizer metrics are z-scored; for notational simplicity, we use $x(T, s)$, $x(T, t)$, and $p(T, s, t)$ to denote these standardized values. We then estimate the pair-level term $u_{s\to t}$ jointly with this tokenizer offset using a mixed-effects regression. Because $\text{BTS}^T_{s\to t}$ is bounded above by 2 and skewed, we model a transformation $y^T_{s\to t} = g(\text{BTS}^T_{s\to t})$ rather than the raw score.[5] We write $y^T_{s\to t}$ as the per-pair term $u_{s\to t}$, modeled as a random effect across language pairs, plus a tokenizer offset linear in $x(T, s)$, $x(T, t)$ and $p(T, s, t)$, plus residual noise ε:

$$g(\text{BTS}^T_{s\to t}) = \beta_0 + \beta_s\, x(T, s) + \beta_t\, x(T, t) + \beta_{st}\, p(T, s, t) + u_{s\to t} + \varepsilon \tag{3}$$

$$\varepsilon \sim \mathcal{N}(0, \sigma^2), \qquad u_{s\to t} \sim \mathcal{N}(\mu, \sigma^2_u) \tag{4}$$

In this formulation, we choose to model $x(T, s)$, $x(T, t)$ and $p(T, s, t)$ as fixed effects and $u_{s\to t}$ as a random effect. Simply put, the former are standard terms in a regression model, while the latter is modeled as a variable drawn from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2_u)$ with learned parameters μ, σ^2_u. Modeling $u_{s\to t}$ as a random effect enables regularization towards a pooled mean across the different $u_{s\to t}$. While there are several motivations for this modeling choice, arguably the clearest is because we are operating in the low sample regime, i.e., we only have a few instances (each tokenizer instance) with which to estimate each $u_{s\to t}$ term. See Appendix B.1 for a more in-depth explanation and justification of these modeling choices.

By marginalizing over the *fixed effects* (i.e., the tokenizer-associated variation),[6] the inherent per-pair transfer baseline, which we call the **Tokenizer-adjusted Bilingual Transfer Score (TBTS)**, can be recovered after inverting the transform:

$$\text{TBTS}_{s\to t} = g^{-1}(\hat{\beta}^{\text{marg}}_0 + \hat{u}_{s\to t}). \tag{5}$$

If the intrinsic metrics of the tokenizer T can be measured on (s, t), then $\text{BTS}^T_{s\to t}$ can also be estimated by adding these variables back in to explicitly account for tokenizer effects. We denote this estimate by $\text{TBTS}^T_{s\to t}$:

$$\text{TBTS}^T_{s\to t} = g^{-1}\big(g(\text{TBTS}_{s\to t}) + \hat{\beta}_s\, x(T, s) + \hat{\beta}_t\, x(T, t) + \hat{\beta}_{st}\, p(T, s, t)\big) \tag{6}$$

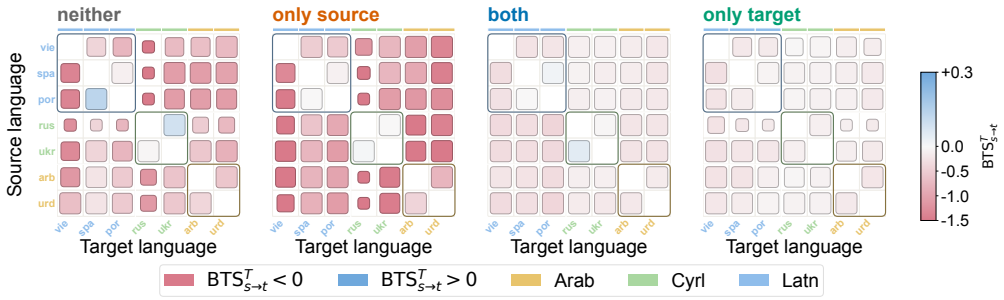

Figure 1: **$BTS^T_{s\to t}$ matrices on different tokenizer conditions.** Each panel shows $BTS^T_{s\to t}$ for directed source–target language pairs under one tokenizer condition: neither, only source, balanced source–target (both), and only target. Rows are source languages, and columns are target languages. Cell color indicates $BTS^T_{s\to t}$, with blue denoting positive transfer and red denoting negative transfer. A larger cell denotes a smaller spread, hence a better reliability.

4 Experimental Setup

Tokenizer Settings To investigate whether $BTS^T_{s\to t}$ varies across different tokenizers, we design five training-coverage settings for each language pair. For each pair (s, t), we train five byte-level BPE tokenizers with a fixed vocabulary: both (source+target, balanced), only source, and only target differ in which of the two languages is seen; neither is trained on all tested languages *except* those written in the same script as s or t, so that neither language's script is represented; and control is an English-only tokenizer held fixed across all pairs as a control. The vocabulary size is held fixed across all pairs so that source- and target-coverage comparisons are not confounded by vocabulary budget. This yields five scores per pair: $BTS^{\text{both}}_{s\to t}$, $BTS^{\text{only source}}_{s\to t}$, $BTS^{\text{only target}}_{s\to t}$, $BTS^{\text{neither}}_{s\to t}$, and $BTS^{\text{control}}_{s\to t}$.

Languages and data We use language pairs spanning a range of scripts and families (Appendix C). Non-English data is drawn from FineWeb2 (Penedo et al., 2025), and English data from FineWeb (Penedo et al., 2024), both used for tokenizer and model training.

$BTS^T_{s\to t}$ measurement. For each tokenizer, we follow the $BTS^T_{s\to t}$ estimation protocol of Longpre et al. (2026), given in Section 3. Concretely, we train a *monolingual* model on the target language t and a *bilingual* model on a balanced (50% s, 50% t) mixture,[7] using the monolingual and bilingual target-loss curves for $BTS^T_{s\to t}$ computations. All models use the Apertus architecture (Apertus et al., 2025) at the 500M scale (Appendix D). As a stability analysis, we recompute $BTS^T_{s\to t}$ over a set of nearby reference points, including 0.8, 0.85, 0.9, and $0.95b_{mono}$. We look at the spread: the difference between the maximum and minimum $BTS^T_{s\to t}$ among them. Large spread indicates a $BTS^T_{s\to t}$ sensitive to the exact reference point and should be interpreted with caution. The recomputed $BTS^T_{s\to t}$ values are not used in the analysis, but the spread is presented in figures as a diagnostic of reliability.

5 Understanding the Variation of $BTS^T_{s\to t}$

Condition-specific $BTS^T_{s\to t}$ matrices mainly show two patterns. For a subset of 7 languages , we computed the $BTS^T_{s\to t}$ under four tokenizer conditions for every ordered pair of distinct languages

[5] $BTS^T_{s\to t}$ is upper-bounded by 2 and empirically exhibits a skewed distribution, so a Gaussian model fit to the raw score can produce fitted values above the bound and skewed residuals. The transformation $y^T_{s\to t} = g(BTS^T_{s\to t}) = -\log(1 - BTS^T_{s\to t}/2)$ maps $BTS^T_{s\to t}$ to $(-\infty, \infty)$. We show the graph of g and the distribution of $y^T_{s\to t}$ in Appendix B.2.

[6] Because $x(T, \cdot)$ and $p(T, s, t)$ are z-scored in the regression, marginalizing over these variables introduces a constant offset on the transformed scale. We absorb this offset into the intercept $\hat{\beta}_0^{\text{marg}}$ used in Eq. 5; this intercept is therefore not the same as the regression intercept β_0 in Eq. 3.

[7] The bilingual corpus is balanced in terms of tokens.

(s, t).[8] Figure 1 shows the matrices for all language pairs under each of the four conditions. The first critical observation is that the matrices are far from identical across conditions. Ideally, these values across conditions would be the same (i.e., independent of the tokenizer choice), but instead we observe substantial shifts in $\text{BTS}^T_{s\to t}$. Interestingly, the first two matrices under neither and only source conditions are relatively similar, and the last two, under only target and both conditions, also show another pattern. This distinction suggests that whether the target language is included in tokenizer training has a big influence on $\text{BTS}^T_{s\to t}$.

Target-side coverage shifts $\text{BTS}^T_{s\to t}$ more than source-side coverage. We now ask *how* $\text{BTS}^T_{s\to t}$ varies with the choice of tokenizer training languages. In Figure 2, we more directly compare the four coverage conditions (both, only source, only target, and neither). We see that source and target coverage are associated with $\text{BTS}^T_{s\to t}$ in different ways. When we consider whether the target language is included in tokenizer training (compare both and only source), the overall distribution of $\text{BTS}^T_{s\to t}$ under both is relatively more concentrated than that of only source, and its median is substantially higher. Low-transfer pairs exhibit the largest score shift between these two conditions, compared to those with high scores. By contrast, the conditions that differ by whether the source language is included in tokenizer training (only target vs. both) show only a slight shift, indicating an opposite but non-significant association. This figure is consistent with the previous paragraph: We see the largest differences in $\text{BTS}^T_{s\to t}$ as a function of target-language coverage, whereas for source coverage we see only marginal differences. These differences are not just a measurement artifact of $\text{BTS}^T_{s\to t}$; they can also be seen early in training. We provide a supplementary analysis in Appendix F.

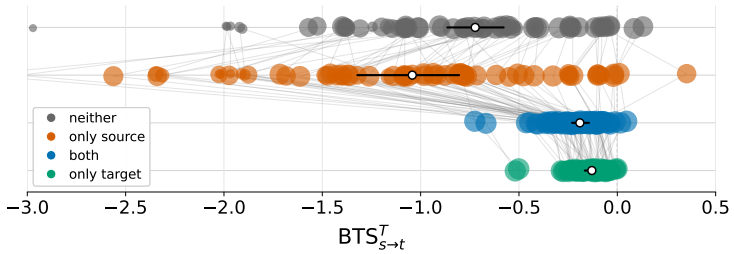

Figure 2: **Distribution of $\text{BTS}^T_{s\to t}$ by tokenizer condition.** Each point is a language pair under a tokenizer condition. Points are grouped by condition (only source, both, and only target, and neither), and adjacent points from the same directed pair are connected with lines. Larger bubbles indicate more stable, reliable estimates, defined by a smaller spread (Section 4). Black intervals show bootstrap 95% CI for the median BTS within each condition.

5.1 Tokenizer–Language Interactions

How much $\text{BTS}^T_{s\to t}$ varies across tokenizers is also dependent on the linguistic relationship between the two languages.

The coverage shift differs depending on source and target language script. Figure 3 fixes a single controlled tokenizer change, the contrast between both and only source. In the left sub-figure, we see that under both, within-script and cross-script pairs have nearly overlapping $\text{BTS}^T_{s\to t}$ distributions (leftmost points). However, once the target language is dropped (only source) from tokenizer training, the two separate, with the $\text{BTS}^T_{s\to t}$ estimates for cross-script pairs falling more sharply. The coverage association presented in Section 5 is thus concentrated among cross-script pairs. The right sub-figure shows that among cross-script pairs alone, the size of this drop varies with the target script.

[8] We analyze the asymmetry between the two transfer directions in Appendix E.

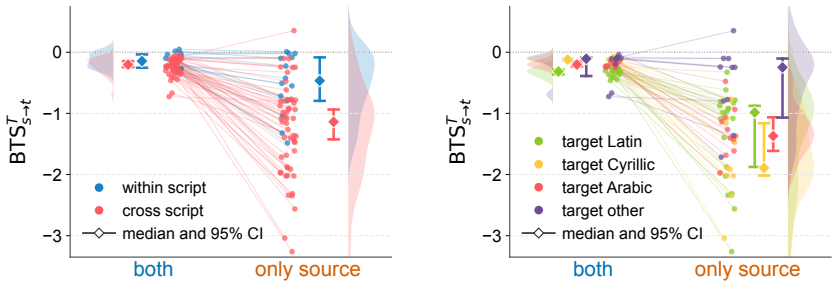

Figure 3: **Script-dependent shift between only source and both conditions.** The left panel compares within-script and cross-script pairs, while the right panel restricts to cross-script pairs and groups them by target script.

Sensitivity of a language pair's $BTS^T_{s\to t}$ estimate to tokenizer choice is associated with average pair transfer strength. Figure 4 plots each directed pair's standard deviation of its $BTS^T_{s\to t}$ across the four tokenizer conditions against its mean $BTS^T_{s\to t}$. Holding the set of tokenizer conditions fixed, sensitivity is associated with average pair transfer strength (i.e., mean $BTS^T_{s\to t}$). High mean-$BTS^T_{s\to t}$ pairs vary little across conditions, whereas low or negative mean-$BTS^T_{s\to t}$ pairs tend to be the most tokenizer-sensitive. A natural interpretation is that high-transfer pairs share enough structure such that they are segmented compatibly by either language's tokenizer, and so show little variation in $BTS^T_{s\to t}$ across tokenizers. We perform a similarly motivated analysis, examining variation in $BTS^T_{s\to t}$ as a function of $TBTS_{s\to t}$, and reach a similar conclusion (see Appendix G).

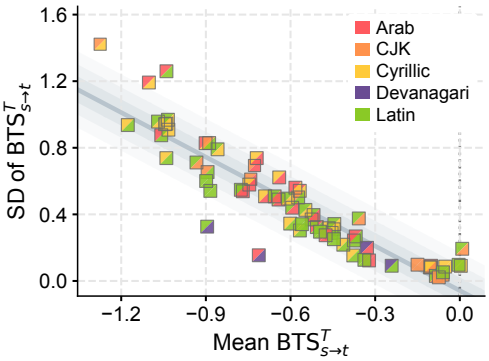

Figure 4: **Sensitivity of $BTS^T_{s\to t}$ estimates as a function of mean $BTS^T_{s\to t}$.** Square markers: the upper-left and the lower-right triangles show the source and target script, respectively. Within-script pairs appear as single-color squares.

6 Accounting for Tokenizer Properties in $BTS^T_{s\to t}$

Section 5 provided evidence that the $BTS^T_{s\to t}$ estimate for a given language pair is strongly affected by tokenizer properties. This section reports $TBTS_{s\to t}$, our proposed score that adjusts for tokenizer-associated variation using the mixed-effects regression setup of eq. (3).

6.1 Metric Choice

Source and target, fertility.[9] For the two per-language predictors ($x(T,s)$,$x(T,t)$), we use fertility on the source and target sides, respectively. Under byte-level BPE, fertility also reflects how much of a language appeared in the tokenizer's training corpus: BPE merges byte sequences that are frequent in that corpus into longer tokens, so a language's fertility falls as its share of the corpus rises, from near byte-level segmentation for absent languages to a few tokens per word for well-represented ones. Fertility thus serves as a proxy for the training-coverage signal of Section 5 as a continuous per-language measure.

Pairwise, vocabulary overlap. For the pairwise predictor ($p(T,s,t)$), we use vocabulary overlap. Vocabulary overlap is the distributional similarity of the two languages' vocabularies under T, captur-

[9]We also run a case study using bigram entropy and vocabulary utilization in our regression in Appendix I.

ing their similarity in this tokenizer's representation. It stands for the residual pair-level correlation beyond the per-side properties, consistent with the within- vs. cross-script split in Section 5.1.

6.2 Results

Held-out tokenizer prediction. In order to know whether the regression captures tokenizer-dependent variation, we perform a leave-one-out prediction check. For each tokenizer condition, we fit the mixed-effects model on the other four conditions and predict raw $\text{BTS}^T_{s\to t}$ using eq. (6) for every directed pair under the held-out condition using that tokenizer's measured fertility and overlap. As a baseline predictor, we use the mean of the pair's raw scores over the four training conditions (excluding the held-out one). We use mean absolute error to evaluate the accuracy of prediction, which is defined as the average of the absolute difference between prediction and observed raw $\text{BTS}^T_{s\to t}$ before the g transformation, across all language pairs. We show the comparison between the regression and the baseline in Figure 5.

Across all held-out conditions, the regression prediction has lower mean absolute error than the per-pair mean baseline on the predicted $\text{BTS}^T_{s\to t}$. The average MAE gap between the regression (0.254) and the baseline (0.471) is considerably large for only source and only target. This indicates that the fitted tokenizer metrics capture systematic, reusable variation across tokenizers from tokenizer intrinsic metrics (fertility and vocabulary overlap), rather than re-estimating a pair-level average.

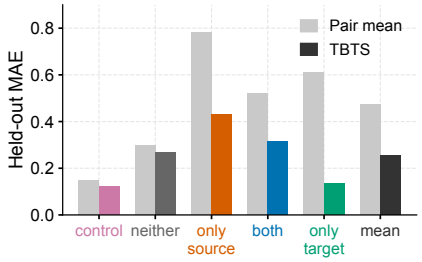

Figure 5: **Leave-one-condition-out prediction error (MAE).**

Table 1: Mixed-effects regression estimates.

Parameter	Estimate	SE	95% CI
Fixed effects			
Intercept β_0^{marg}	-0.114	0.022	$[-0.157, -0.071]$
Target γ	-0.639	0.032	$[-0.701, -0.577]$
Source δ	$+0.199$	0.031	$[+0.139, +0.260]$
Overlap η	$+0.296$	0.061	$[+0.177, +0.414]$
$\hat{u}_{s\to t}$ SD	0.109	—	$[0.084, 0.132]$
Residual SD	0.119	—	$[0.108, 0.129]$

Final Formula of $\text{TBTS}^T_{s\to t}$. Given the accuracy of the regression in held-out prediction, we fit eq. (3) with the three predictors above on the data from all five tokenizer conditions. All three fixed effects are statistically significant, i.e., larger than would plausibly arise from noise alone under the model (Table 1). The fitted coefficients give the following $\text{TBTS}^T_{s\to t}$:

$$\boxed{g^{-1}\left(\beta_0^{\text{marg}} + \gamma F_{t,T} + \delta F_{s,T} + \eta O_{s,t,T} + \hat{u}_{s\to t}\right)} \tag{7}$$

Here, $F_{t,T}$ and $F_{s,T}$ denote the fertility of tokenizer T on the target and source languages, and $O_{s,t,T}$ denotes the vocabulary overlap for pair $s \to t$ under T; all three are the z-scored values used in the regression. Table 1 reports the estimates with standard errors and 95% confidence intervals. Higher target fertility is associated with lower $\text{BTS}^T_{s\to t}$, while higher source fertility and vocabulary overlap are associated with higher $\text{BTS}^T_{s\to t}$. Because the predictors are standardized, the coefficient magnitudes are directly comparable: the target-side coefficient ($|\hat{\gamma}| = 0.639$) is larger than the source-side coefficient ($|\hat{\delta}| = 0.199$). This again indicates that target-language coverage in tokenizer training data is a large factor in $\text{BTS}^T_{s\to t}$ estimates.

Tokenizer adjustment changes both the values and the ordering of language-pair scores relative to the control tokenizer. Appendix J presents the full matrices and pair-level examples.

6.3 Evaluation

Given the fitted results, we evaluate the accuracy of eq. (7) on a realistic multilingual tokenizer, which is trained on evenly sampled 15 languages (English + languages in Table 3). The tokenizer is trained using BPE, with the same vocabulary size and size of data (2.0 GB) as other tokenizers for fairness.

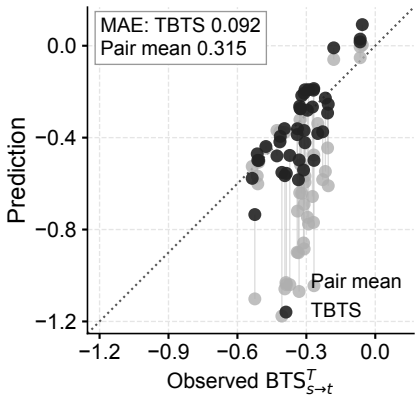

Figure 6: **Multilingual-tokenizer prediction.** Regression (eq. (7)) and pair-mean predictions (y) versus observed $\text{BTS}^T_{s\to t}$ (x).

Table 2: **Intrinsic-metric ablation.** $\neg X$ omits X; the last three metric rows fit one metric. Best values are bold. r: Pearson; τ: Kendall.

Metrics	MAE ↓	RMSE ↓	r ↑	τ ↑
All	0.092	**0.105**	**0.817**	0.561
$\neg O_{s,t,T}$	0.101	0.122	0.717	0.540
$\neg F_{s,T}$	**0.086**	0.114	0.806	**0.570**
$\neg F_{t,T}$	0.294	0.317	0.639	0.247
$F_{t,T}$	0.092	0.122	0.760	0.489
$F_{s,T}$	0.299	0.316	0.552	0.306
$O_{s,t,T}$	0.239	0.264	0.618	0.180
Pair-mean	0.315	0.400	0.436	0.222

Besides MAE, we also evaluate the RMSE (Root Mean Square Error) and Pearson r between the predicted raw $\text{BTS}^T_{s\to t}$ and the ground truth. Because the ranking of scores across different language pairs is also important in comparing transfer efficiency, we additionally use Kendall's τ to measure rank similarity.

Figure 6 shows the results of regression prediction and pair-mean baseline. We notice the pair-mean baseline provides a more biased prediction compared to our regression, underestimating the $\text{BTS}^T_{s\to t}$ for a number of pairs. Regarding MAE, the regression also yields a better result (MAE = 0.092) than the baseline method (MAE = 0.315).

7 Ablation Studies

Ablation studies of intrinsic metrics. To assess the contribution of each metric involved, we conduct an ablation study by removing certain metrics. The results in Table 2 confirm that target-side fertility is the most crucial metric, without which the regression prediction is similar to the pair-mean baseline. Removing overlap consistently causes a performance drop on four columns, suggesting vocabulary overlap provides marginal but effective signals. The contribution of source-side fertility is less consistent, as removing it improves performance under MAE and Kendall's τ. However, since our multilingual-tokenizer evaluation involves only one tokenizer, source-side fertility, obtained without additional computation, may become informative in other cases, so we keep it in the regression.

Ablation studies of tokenizer conditions. We also conduct another ablation study by removing data from a certain tokenizer condition and show the results in Appendix K. This ablation study suggests that data from different tokenizers may help generalization to unknown realistic tokenizers.

8 Conclusion

In this paper, we hypothesize that Bilingual Transfer Score estimates are confounded by tokenizer choice. Our analyses across different tokenizer settings indeed show that $\text{BTS}^T_{s\to t}$ estimates vary greatly depending on the underlying tokenizer choice, evincing that the original $\text{BTS}^T_{s\to t}$ estimates reflect not just the level of directional language transfer efficiency, but also tokenizer artifacts. Further, the degree of this dependency on the tokenizer appears entangled with language-pair properties, such as script profiles of the pair. To adjust for these tokenizer artifacts, we quantify tokenizer properties with intrinsic metrics and fit a mixed-effects regression that separates pair-level transfer from tokenizer-associated variation. Fit on the five coverage conditions, the model predicts $\text{BTS}^T_{s\to t}$ under a held-out multilingual tokenizer with a mean absolute error of 0.092, against 0.315 for a per-pair mean baseline, without training any model under that tokenizer. The resulting $\text{TBTS}_{s\to t}$ is thus less tied to the tokenizer used in estimation. Limitations are discussed in Appendix L.

AI USE STATEMENT

We used generative AI tools to assist with writing and polishing the manuscript, including grammar checking and improving clarity. AI assistance was also used to improve the LaTeX formatting of figures and tables. The authors take full responsibility for the accuracy and integrity of the final manuscript.

REPRODUCIBILITY STATEMENT

To support reproducibility, we specify the data sources, tokenizer conditions, and BTS estimation procedure in Section 4, and provide language coverage and training details in Appendices C and D. Tokenizer metrics are defined in Appendix A. Sections 3.2 and 6 describe the regression model and the evaluation of predictions.

REFERENCES

Diana Abagyan, Alejandro R. Salamanca, Andres Felipe Cruz-Salinas, Kris Cao, Hangyu Lin, Acyr Locatelli, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. One tokenizer to rule them all: Emergent language plasticity via multilingual tokenizers, 2025. URL https://arxiv.org/abs/2506.10766.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. Do all languages cost the same? tokenization in the era of commercial language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9904–9923, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.614. URL https://aclanthology.org/2023.emnlp-main.614/.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Valentin Hofmann, Tomasz Limisiewicz, Yulia Tsvetkov, and Noah A. Smith. Magnet: Improving the multilingual fairness of language models with adaptive gradient-based tokenization, 2024. URL https://arxiv.org/abs/2407.08818.

Belen Alastruey, João Maria Janeiro, Alexandre Allauzen, Maha Elbayad, Loïc Barrault, and Marta R. Costa-jussà. Interference matrix: Quantifying cross-lingual interference in transformer encoders, 2025. URL https://arxiv.org/abs/2508.02256.

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, Stefan Kesselheim, and Nicolas Flores-Herr. Tokenizer choice for LLM training: Negligible or crucial? In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3907–3924, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.247. URL https://aclanthology.org/2024.findings-naacl.247/.

Project Apertus, Alejandro Hernández-Cano, Alexander Hägele, Allen Hao Huang, Angelika Romanou, Antoni-Joan Solergibert, Barna Pasztor, Bettina Messmer, Dhia Garbaya, Eduard Frank Ďurech, Ido Hakimi, Juan García Giraldo, Mete Ismayilzada, Negar Foroutan, Skander Moalla, Tiancheng Chen, Vinko Sabolčec, Yixuan Xu, Michael Aerni, Badr AlKhamissi, Inés Altemir Mariñas, Mohammad Hossein Amani, Matin Ansaripour, Ilia Badanin, Harold Benoit, Emanuela Boros, Nicholas Browning, Fabian Bösch, Maximilian Böther, Niklas Canova, Camille Challier, Clement Charmillot, Jonathan Coles, Jan Deriu, Arnout Devos, Lukas Drescher, Daniil Dzenhaliou, Maud Ehrmann, Dongyang Fan, Simin Fan, Silin Gao, Miguel Gila, María Grandury, Diba Hashemi, Alexander Hoyle, Jiaming Jiang, Mark Klein, Andrei Kucharavy, Anastasiia Kucherenko, Frederike Lübeck, Roman Machacek, Theofilos Manitaras, Andreas Marfurt, Kyle Matoba, Simon Matrenok, Henrique Mendonça, Fawzi Roberto Mohamed, Syrielle Montariol, Luca Mouchel, Sven Najem-Meyer, Jingwei Ni, Gennaro Oliva, Matteo Pagliardini, Elia Palme, Andrei Panferov, Léo Paoletti, Marco Passerini, Ivan Pavlov, Auguste Poiroux, Kaustubh Ponkshe, Nathan Ranchin, Javi Rando, Mathieu Sauser, Jakhongir Saydaliev, Muhammad Ali Sayfiddinov, Marian Schneider, Stefano Schuppli, Marco Scialanga, Andrei Semenov, Kumar Shridhar, Raghav Singhal, Anna Sotnikova, Alexander Sternfeld, Ayush Kumar Tarun, Paul Teiletche, Jannis Vamvas, Xiaozhe Yao, Hao Zhao, Alexander Ilic, Ana Klimovic, Andreas Krause, Caglar Gulcehre, David Rosenthal, Elliott Ash, Florian Tramèr, Joost VandeVondele, Livio Veraldi, Martin Rajman, Thomas Schulthess, Torsten Hoefler, Antoine Bosselut, Martin Jaggi, and Imanol Schlag. Apertus: Democratizing open and compliant llms for global language environments, 2025. URL https://arxiv.org/abs/2509.14233.

Catherine Arnett and Benjamin Bergen. Why do language models perform worse for morphologically complex languages? In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 6607–6623, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.441/.

Catherine Arnett, Marisa Hudspeth, and Brendan O'Connor. Evaluating morphological alignment of tokenizers in 70 languages. In *Tokenization Workshop*, 2026. URL https://openreview.net/forum?id=XYRri1s6pP.

Lisa Beinborn and Yuval Pinter. Analyzing cognitive plausibility of subword tokenization. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4478–4486, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.272. URL https://aclanthology.org/2023.emnlp-main.272/.

Verena Blaschke, Masha Fedzechkina, and Maartje Ter Hoeve. Analyzing the effect of linguistic similarity on cross-lingual transfer: Tasks and experimental setups matter. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 8653–8684, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.454. URL https://aclanthology.org/2025.findings-acl.454/.

Terra Blevins, Hila Gonen, and Luke Zettlemoyer. Analyzing the mono- and cross-lingual pretraining dynamics of multilingual language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3575–3590, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.234. URL https://aclanthology.org/2022.emnlp-main.234/.

Marco Cognetta, Vilém Zouhar, Sangwhan Moon, and Naoaki Okazaki. Two counterexamples to tokenization and the noiseless channel. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 16897–16906, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1469/.

Gautier Dagan, Gabriel Synnaeve, and Baptiste Rozière. Getting the most out of your tokenizer for pre-training and domain adaptation, 2024. URL https://arxiv.org/abs/2402.01035.

Mete Erdogan, Abhiram Gorle, Shubham Chandak, Mert Pilanci, and Tsachy Weissman. An information-theoretic perspective on llm tokenizers, 2026. URL https://arxiv.org/abs/2601.09039.

Negar Foroutan, Clara Meister, Debjit Paul, Joel Niklaus, Sina Ahmadi, Antoine Bosselut, and Rico Sennrich. Parity-aware byte-pair encoding: Improving cross-lingual fairness in tokenization, 2025. URL https://arxiv.org/abs/2508.04796.

Omer Goldman, Avi Caciularu, Matan Eyal, Kris Cao, Idan Szpektor, and Reut Tsarfaty. Unpacking tokenization: Evaluating text compression and its correlation with model performance. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2274–2286, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.134. URL https://aclanthology.org/2024.findings-acl.134/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet,

Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia

Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Katharina Hämmerl, Tomasz Limisiewicz, Jindřich Libovický, and Alexander Fraser. Beyond literal token overlap: Token alignability for multilinguality. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 756–767, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-190-2. doi: 10.18653/v1/2025.naacl-short.63. URL https://aclanthology.org/2025.naacl-short.63/.

Yifei He, Alon Benhaim, Barun Patra, Praneetha Vaddamanu, Sanchit Ahuja, Parul Chopra, Vishrav Chaudhary, Han Zhao, and Xia Song. Scaling laws for multilingual language models, 2024. URL https://arxiv.org/abs/2410.12883.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 30016–30030, 2022.

Yifan Hu, Frank Liang, Dachuan Zhao, Jonathan Geuter, Varshini Reddy, Craig W. Schmidt, and Chris Tanner. Entropy-driven pre-tokenization for byte-pair encoding, 2025. URL https://arxiv.org/abs/2506.15889.

Julie Kallini, Dan Jurafsky, Christopher Potts, and Martijn Bartelds. False Friends are not foes: Investigating vocabulary overlap in multilingual language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 21138–21154, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1153. URL https://aclanthology.org/2025.findings-emnlp.1153/.

Vani Kanjirangat, Tanja Samardzic, Ljiljana Dolamic, and Fabio Rinaldi. Tokenization and representation biases in multilingual models on dialectal NLP tasks. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 23992–24010, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1224. URL https://aclanthology.org/2025.emnlp-main.1224/.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *CoRR*, abs/2001.08361, 2020. URL https://arxiv.org/pdf/2001.08361.pdf.

Muhammad Dehan Al Kautsar and Fajri Koto. Parallel tokenizers: Rethinking vocabulary design for cross-lingual transfer, 2025. URL https://arxiv.org/abs/2510.06128.

Tomasz Limisiewicz, Jiří Balhar, and David Mareček. Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5661–5681, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.350. URL https://aclanthology.org/2023.findings-acl.350/.

Tomasz Limisiewicz, Terra Blevins, Hila Gonen, Orevaoghene Ahia, and Luke Zettlemoyer. MYTE: Morphology-driven byte encoding for better and fairer multilingual language modeling. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15059–15076, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.804. URL https://aclanthology.org/2024.acl-long.804/.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, and Yejin Choi. Superbpe: Space travel for language models, 2025. URL https://arxiv.org/abs/2503.13423.

Shayne Longpre, Sneha Kudugunta, Niklas Muennighoff, I-Hung Hsu, Isaac Caswell, Alex Pentland, Sercan Arik, Chen-Yu Lee, and Sayna Ebrahimi. ATLAS: Adaptive transfer scaling laws for multilingual pretraining, finetuning, and decoding the curse of multilinguality, 2026. URL https://arxiv.org/abs/2510.22037.

Clara Meister. Tokeval: A tokenizer analysis suite, 2025. URL https://github.com/swiss-ai/tokenizer-intrinsic-evals.

Byung-Doh Oh and William Schuler. The impact of token granularity on the predictive power of language model surprisal. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4150–4162, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.209. URL https://aclanthology.org/2025.acl-long.209/.

Volodymyr Ovcharov. The tokenizer tax across 25 european languages: Domain invariance, cross-lingual few-shot effects, and the ukrainian penalty, 2026. URL https://arxiv.org/abs/2605.24718.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849. Curran Associates, Inc., 2024. doi: 10.52202/079017-0970. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/370df50ccfdf8bde18f8f9c2d9151bda-Paper-Datasets_and_Benchmarks_Track.pdf.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. Fineweb2: One pipeline to scale them all – adapting pre-training data processing to every language, 2025. URL https://arxiv.org/abs/2506.20920.

Aleksandar Petrov, Emanuele La Malfa, Philip H. S. Torr, and Adel Bibi. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.15425.

Wessel Poelman, Thomas Bauwens, and Miryam de Lhoneux. Confounding factors in relating model performance to morphology. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7262–7287, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.369. URL https://aclanthology.org/2025.emnlp-main.369/.

François Remy, Pieter Delobelle, Hayastan Avetisyan, Alfiya Khabibullina, Miryam de Lhoneux, and Thomas Demeester. Trans-tokenization and cross-lingual vocabulary transfers: Language adaptation of llms for low-resource nlp, 2024. URL https://arxiv.org/abs/2408.04303.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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)*, pp. 3118–3135, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.243. URL https://aclanthology.org/2021.acl-long.243/.

Haruki Sakajo, Yusuke Ide, Justin Vasselli, Yusuke Sakai, Yingtao Tian, Hidetaka Kamigaito, and Taro Watanabe. Dictionaries to the rescue: Cross-lingual vocabulary transfer for low-resource languages using bilingual dictionaries. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25963–25976, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1333. URL https://aclanthology.org/2025.findings-acl.1333/.

Craig W Schmidt, Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. Tokenization is more than compression. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 678–702, 2024.

Mir Tafseer Nayeem, Sawsan Alqahtani, Md Tahmid Rahman Laskar, Tasnim Mohiuddin, and M Saiful Bari. Beyond fertility: Analyzing strr as a metric for multilingual tokenization evaluation. *arXiv e-prints*, pp. arXiv–2510, 2025.

Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. Scaling laws with vocabulary: Larger models deserve larger vocabularies. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=sKCKPr8cRL.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Menan Velayuthan and Kengatharaiyer Sarveswaran. Egalitarian language representation in language models: It all begins with tokenizers. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5987–5996, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.400/.

Chen Zhang, Zhiyuan Liao, and Yansong Feng. Cross-lingual transfer of cultural knowledge: An asymmetric phenomenon. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 147–157, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-252-7. doi: 10.18653/v1/2025.acl-short.13. URL https://aclanthology.org/2025.acl-short.13/.

Vilém Zouhar, Clara Meister, Juan Gastaldi, Li Du, Mrinmaya Sachan, and Ryan Cotterell. Tokenization and the noiseless channel. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5184–5207, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.284. URL https://aclanthology.org/2023.acl-long.284/.

A Tokenizer Intrinsic Metrics: Definitions

All intrinsic metrics are computed per language on a held-out evaluation corpus using the `tokenizer-analysis` library Meister (2025). Let $\mathcal{S}$ denote a tokenizer, l a language, C_l a

corpus of texts in language l, and $T_l = \{t_1, t_2, \ldots\}$ the sequence of token ids produced by tokenizing C_l with $\mathcal{S}$. We write $V_l = \{t : t \in T_l\}$ for the set of distinct token types observed in C_l, and $p_l(t) = \text{count}(t, T_l)/|T_l|$ for the empirical unigram probability of token t in C_l.

PER-LANGUAGE METRICS

Fertility. Average number of tokens produced per word:

$$\text{fertility}(\mathcal{S}, l) = \frac{1}{|D_l|} \sum_{d \in D_l} \frac{|\mathcal{S}(d)|}{\text{words}(d)}$$

where D_l is the set of documents in C_l, $\mathcal{S}(d)$ is the token sequence for document d, and $\text{words}(d)$ is its word count. Higher fertility indicates a finer granularity of segmentation.

Compression Rate. Ratio-of-means of characters to tokens across the corpus:

$$\text{CR}(\mathcal{S}, l) = \frac{\sum_{d \in D_l} \text{chars}(d)}{\sum_{d \in D_l} |\mathcal{S}(d)|}$$

Higher compression rate means fewer tokens are needed to represent the text. Compression rate and fertility are inversely related but differ in normalisation: fertility is the mean of per-document ratios, while compression rate is the ratio of corpus-level totals, which reduces the influence of short texts.

Rényi Entropy. The order-α Rényi entropy of the unigram token distribution in language l:

$$H_\alpha(\mathcal{S}, l) = \frac{1}{1-\alpha} \log_2\Big(\sum_{t \in V_l} p_l(t)^\alpha \Big), \ \alpha \neq 1$$

At $\alpha \to 1$ this converges to the Shannon entropy

$$H_1(\mathcal{S}, l) = -\sum_t p_l(t) \log_2 p_l(t)$$

(unigram entropy below). For $\alpha > 1$, H_α places increasing weight on high-frequency tokens, making it sensitive to whether the distribution is dominated by a few types. We evaluate at $\alpha \in \{1.0, 2.0, 2.5, 3.0\}$. The metric is also referred to as Rényi entropy in our analyses, as it reflects how evenly the vocabulary is utilised.

Unigram Entropy. Shannon entropy of the unigram token distribution:

$$H_1(\mathcal{S}, l) = -\sum_{t \in V_l} p_l(t) \log_2 p_l(t)$$

Higher entropy indicates a more uniform token distribution, hence more efficient use of the vocabulary for that language.

Bigram Entropy. Following Poelman et al. (2025), we measure the normalised Shannon entropy of right-successor distributions (denoted η in the original work). For each token type t occurring as the left element of at least $k_{\min} = 3$ bigrams, let $\text{succ}(t)$ denote the multiset of its right-successors and $n_t = |\text{succ}(t)|$.

The per-type normalised entropy is:

$$\eta_t = \frac{H(\text{succ}(t))}{\log_2 \min(|\{s : (t, s)\}|, n_t)}$$

where $H(\cdot)$ is Shannon entropy. The corpus-level bigram entropy is:

$$\mathrm{BigramH}(\mathcal{S}, l) = \frac{\sum_{t:n_t \geq k_{\min}} n_t \eta_t}{\sum_{t:n_t \geq k_{\min}} n_t}$$

Higher bigram entropy indicates more uniform successor distributions, reflecting less predictable token sequences and, by extension, more compositional segmentation.

Vocabulary Utilization. Fraction of the full vocabulary $V_{\mathcal{S}}$ that is observed in C_l:

$$\mathrm{VocabUtil}(\mathcal{S}, l) = \frac{|V_l|}{|V_{\mathcal{S}}|} \in [0, 1]$$

UTF-8 Integrity (Token Completeness Rate). A token is *UTF-8 complete* if the bytes it represents form one or more valid, complete UTF-8 characters. The integrity rate is the fraction of content tokens that are UTF-8 complete:

$$\mathrm{UTF8Int}(\mathcal{S}, l) = \frac{|\{t_i \in T_l : \mathrm{UTF8complete}(t_i)\}|}{|T_l|}$$

with values in $[0, 1]$. A value below 1 indicates that the tokenizer splits multi-byte characters across token boundaries (common in byte-level BPE tokenizers applied to non-Latin scripts).

UTF-8 Char Split (Splits per 1 000 Tokens). Let n_{split} be the number of multi-byte Unicode characters in C_l whose bytes are distributed across two or more tokens. The metric normalises this by total token count:

$$\mathrm{UTF8Split}(\mathcal{S}, l) = \frac{n_{\mathrm{split}}}{|T_l|} \times 10^3$$

It is computed via byte-stream alignment of the tokenised output against the original UTF-8 source bytes. High values indicate frequent character boundary violations.

PAIRWISE VOCABULARY OVERLAP METRICS

Let $\mathcal{T} = \{l_1, \ldots, l_K\}$ be the set of languages used to train tokenizer $\mathcal{S}$.

$$\mathrm{overlap}(l_1, l_2) = \frac{|V_{l_1}^{\mathcal{S}} \cap V_{l_2}^{\mathcal{S}}|}{|V_{l_1}^{\mathcal{S}} \cup V_{l_2}^{\mathcal{S}}|}$$

$$\mathrm{shared_types}(l_1, l_2) = |V_{l_1}^{\mathcal{S}} \cap V_{l_2}^{\mathcal{S}}|$$

$$\%\mathrm{share}_{\mathrm{train}}(l_1, l_2) = \frac{|V_{l_1}^{\mathcal{S}} \cap V_{l_2}^{\mathcal{S}}|}{|V_{l_1}^{\mathcal{S}}|}$$

$$\%\mathrm{share}_{\mathrm{ft}}(l_1, l_2) = \frac{|V_{l_1}^{\mathcal{S}} \cap V_{l_2}^{\mathcal{S}}|}{|V_{l_2}^{\mathcal{S}}|}$$

$$\mathrm{tokens}(l_1) = |V_{l_1}^{\mathcal{S}}|, \quad \mathrm{tokens}(l_2) = |V_{l_2}^{\mathcal{S}}|$$

$$\mathrm{JSD}(l_1, l_2) = \tfrac{1}{2} D_{\mathrm{KL}}(p_{l_1} \| m) + \tfrac{1}{2} D_{\mathrm{KL}}(p_{l_2} \| m), \quad m = \tfrac{1}{2}(p_{l_1} + p_{l_2})$$

B TBTS$_{s\to t}$ FORMULATION

B.1 MIXED-EFFECTS MODELING

This appendix expands on the choice to model the tokenizer metrics $x(T, s)$, $x(T, t)$, and $p(T, s, t)$ as fixed effects and the pair-level term $u_{s\to t}$ as a random intercept.

Two components over a grouped data structure. Each language pair (s, t) is measured under several tokenizers T, so the observations are grouped by pair, with few observations per pair. We separate each $\mathrm{BTS}^T_{s\to t}$ into a pair-level baseline and a tokenizer-associated component. The random intercept $u_{s\to t}$ absorbs the intrinsic transfer level of the pair: the part of the score that is shared across the tokenizers we apply to that pair. The three metrics account for the part that changes with the tokenizer. Estimating the two together requires a model that represents both the per-pair grouping and the metric coefficients.

Fixed effects for the metrics, a random intercept for the pair. We treat the tokenizer-associated component as fixed effects, since we aim to estimate how tokenizer properties are systematically associated with $\mathrm{BTS}^T_{s\to t}$. A fixed effect is a single coefficient, shared by all pairs, that we estimate and report; the sign and size of β_s, β_t, and β_{st} are the object of the analysis. The inherent transfer component is instead a per-pair baseline, of which there are as many as there are pairs. We therefore introduce a language-pair random intercept $u_{s\to t}$, assuming that these pair-specific baselines are drawn from a common Gaussian distribution $u_{s\to t} \sim \mathcal{N}(\mu, \sigma_u^2)$, which enables pooling across language pairs. The parameters μ and σ_u^2 are estimated from all pairs jointly.

B.2 TRANSFORM OF $\mathrm{BTS}^T_{s\to t}$

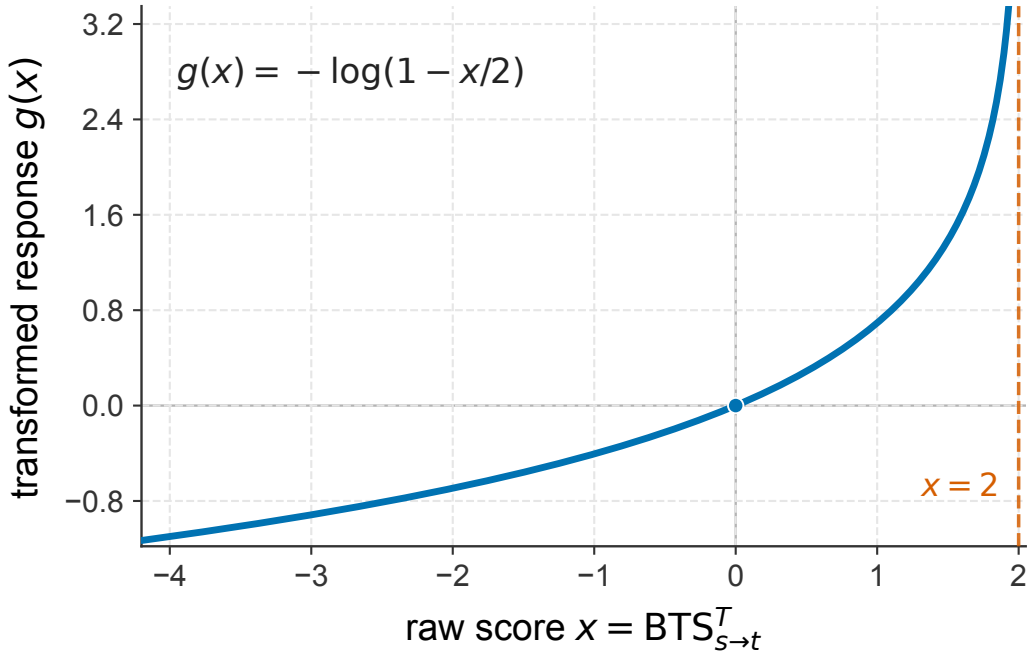

Figure 7: **Monotone response transform for TBTS$^T_{s\to t}$.** The transformation $g(x) = -\log(1 - x/2)$ maps the bounded raw score $x = \mathrm{BTS}^T_{s\to t} \in (-\infty, 2)$ to an unconstrained real-valued response. The dashed vertical line marks the upper boundary $x = 2$. The transform is monotone and satisfies $g(0) = 0$, so it preserves the ordering and sign of transfer scores.

Response transformation. Because $\mathrm{BTS}^T_{s\to t}$ is upper-bounded by 2, we apply the monotone map $g(x) = -\log(1 - x/2)$ before fitting the mixed-effects regression. As shown in Figure 7, this

transformation maps the full domain $(-\infty, 2)$ onto the real line while preserving the ordering of transfer scores. It also keeps zero fixed, so positive and negative transfer remain separated at the same reference point. The regression is fit on $y^T_{s\to t} = g(\mathrm{BTS}^T_{s\to t})$; fitted values can be mapped back to the original scale by $g^{-1}(y) = 2(1 - \exp(-y))$. Figure 8 shows the distribution of $\mathrm{BTS}^T_{s\to t}$ under all four tokenizer conditions before and after the transform, the latter less skewed.

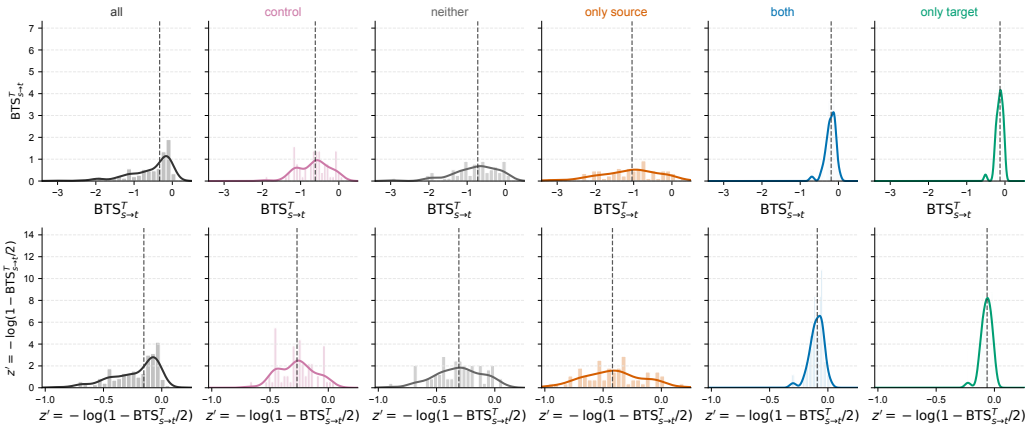

Figure 8: **Distribution of tokenizer-dependent transfer estimates and transformed response.** Columns show the pooled distribution and the four tokenizer conditions. The top row shows the raw $\mathrm{BTS}^T_{s\to t}$ distribution, and the bottom row shows the transformed response $z' = -\log(1-\mathrm{BTS}^T_{s\to t}/2)$ used in the mixed-effects regression. Dashed vertical lines mark condition-specific medians.

C Languages and Experimental Scale

Languages. We study 14 languages spanning 7 scripts (Table 3).

Table 3: Evaluated languages, grouped by script.

Script	Languages
Latin (Latn)	French (fra), Polish (pol), Portuguese (por), Spanish (spa), Turkish (tur), Vietnamese (vie)
Cyrillic (Cyrl)	Russian (rus), Ukrainian (ukr)
Arabic (Arab)	Arabic (arb), Urdu (urd)
CJK	Chinese (cmn), Japanese (jpn), Korean (kor)
Devanagari (Deva)	Hindi (hin)

Language pairs. We evaluate 35 unordered language pairs (70 directed source–target pairs), listed in Table 4.

Tokenizer conditions and scale. For each pair we consider the four coverage conditions (both, only source, only target, neither) together with the shared control tokenizer. Because both is pair-specific while the others are shared across pairs—only source/only target reuse a single only-X tokenizer per language, neither is shared across pairs with the same script combination, and control is a single English-only tokenizer—the five conditions over 35 pairs are realized by 64 independently trained tokenizers: 35 both, 14 per-language only-X, 14 neither, and 1 control.

Table 4: The 35 unordered language pairs. Red marks the 21 pairs forming the complete matrix among Vietnamese, Russian, Ukrainian, Spanish, Portuguese, Arabic, and Urdu.

#	Pair	#	Pair	#	Pair
1	arb–cmn	13	cmn–urd	25	por–vie
2	arb–por	14	fra–jpn	26	rus–spa
3	arb–rus	15	fra–rus	27	rus–ukr
4	arb–spa	16	fra–tur	28	rus–urd
5	arb–ukr	17	hin–spa	29	rus–vie
6	arb–urd	18	hin–urd	30	spa–ukr
7	arb–vie	19	kor–spa	31	spa–urd
8	cmn–jpn	20	pol–rus	32	spa–vie
9	cmn–por	21	por–rus	33	ukr–urd
10	cmn–rus	22	por–spa	34	ukr–vie
11	cmn–spa	23	por–ukr	35	urd–vie
12	cmn–ukr	24	por–urd		

Training runs. Each (pair, condition) defines one bilingual run, while monolingual reference runs are shared across pairs that use the same tokenizer. In total this gives 175 bilingual and 212 monolingual runs (387 runs), broken down in Table 5.

Table 5: Number of model training runs. Monolingual runs are deduplicated across pairs sharing a tokenizer.

Conditions	Bilingual	Monolingual
both + only source + only target	105	154
+ neither	+35	+44
+ control	+35	+14
Total	175	212

D Model and Training Details

Architecture. All models are Llama-style decoder-only Transformers at the 500M scale, summarized in Table 6.

Table 6: Model architecture.

Hyperparameter	Value
Layers	14
Hidden size	1280
FFN intermediate size	7680 (SwiGLU, no bias)
Attention heads	32 query / 8 key–value
Head dimension	40, with QK-norm
Normalization	RMSNorm ($\epsilon = 10^{-5}$)
Positional encoding	RoPE ($\theta = 10^4$)
Max position / training length	4096 / 2048
Embeddings	untied
Non-embedding parameters	$\approx$ 415M (at 64k vocab)
Total parameters	$\approx$ 580M (at 64k vocab)

Optimization. We use AdamW (weight decay 0.1, gradient clipping 1.0) with a constant learning rate after a 200-step warmup, with the peak learning rate set from the model size. Each step uses a micro-batch of 40 sequences with gradient accumulation 2 per language across 4 GPUs, i.e. 320 sequences of length 2048 ($\approx$ 0.66M tokens) per language per step. We compile the model with `torch.compile` and fix the seed to 42.

Token budget. Following Chinchilla-style compute-optimal scaling, each monolingual run is trained to 20$\times$ the non-embedding parameter count in target tokens. Bilingual runs are trained up to 1.5$\times$ the monolingual budget per language, which sets the floor below which $\text{BTS}^{T}_{s\to t}$ is reported without extrapolation.

Tokenizers. All tokenizers are byte-level BPE with a 64k vocabulary, $\text{min_frequency} = 2$, and five special tokens (PAD, BOS, EOS, UNK, MASK). Each tokenizer is trained on 2.0 GB of total text of training languages (uniform weights, shuffle buffer 10000, seed 42), drawn from FineWeb-2 for non-English languages and the English subset of FineWeb (CC-MAIN-2023-50).

E Directionality of Bilingual Transfer

Prior work has shown that cross-lingual transfer in multilingual models can be asymmetric, i.e., transfer from s to t may not be equal to transfer from t to s (Blevins et al., 2022; Zhang et al., 2025).

This is further evinced by the asymmetry of the $\text{BTS}^{T}_{s\to t}$ transfer matrix of ATLAS. We likewise see this asymmetry in the per-condition transfer matrices of Figure 1. To quantify the symmetry of $\text{BTS}^{T}_{s\to t}$, we operationalize the notion of language pair transfer symmetry using what we call the **symmetry ratio**. For each tokenizer condition, we derive a $\text{BTS}^{T}_{s\to t}$ matrix M, where M_{st} is the $\text{BTS}^{T}_{s\to t}$ from source language s to target language t. Then we can decompose M into symmetric and skew-symmetric components and define the symmetry ratio:

$$M_{\text{sym}} = \frac{M + M^{\top}}{2}, \qquad M_{\text{skew}} = \frac{M - M^{\top}}{2}, \qquad \rho_{\text{sym}} = \frac{\|M_{\text{sym}}\|_F}{\sqrt{\|M_{\text{sym}}\|_F^2 + \|M_{\text{skew}}\|_F^2}}. \tag{8}$$

.

Intuitively, this ratio quantifies the proportion of the matrix that can be captured by a purely symmetric matrix, which in turn quantifies generally how equal the BTS estimates for a given language pair are across the two pair directions. The symmetry ratios are consistently high across the four tokenizer conditions: both (0.896) > only target (0.888) > neither (0.879) > only source (0.872). Thus, although individual language pairs can be strongly directional and asymmetric, the dominant component of the $\text{BTS}^{T}_{s\to t}$ matrices is symmetric, which also aligns with Blaschke et al. (2025); Longpre et al. (2026)'s finding that the language transfer results could be overall symmetric.

F Tokenizer affects training dynamics in the BTS estimation procedure

In the hope of better understanding where these differences come from, we compare training dynamics between these conditions. We see that the four tokenizer conditions already diverge during the early stage of model training from which $\text{BTS}^{T}_{s\to t}$ is derived, well before any final score is read off. The neither and the only source conditions start with higher initial validation BPB, while the other two conditions, with target languages covered in the tokenizers, show another decreasing pattern.

Taking a closer look at the only source curves, the Spanish–Portuguese pair starts at a lower validation BPB than the Ukrainian–Portuguese pair, because the tokenizer trained on Spanish and the other tokenizer trained on Ukrainian result in different fragmentation of Portuguese. Even though the training of all four conditions reaches the same level of validation BPB, the Spanish–Portuguese pair gives a higher final $\text{BTS}^{\text{both}}_{s\to t}$ while the Ukrainian–Portuguese pair gives a low negative $\text{BTS}^{\text{both}}_{s\to t}$. Because these differences happen so early in training, we hypothesize that at least some of this difference in $\text{BTS}^{T}_{s\to t}$ can be attributed to how the tokenizer segments the target language, regardless of any *true* transfer abilities between the two.

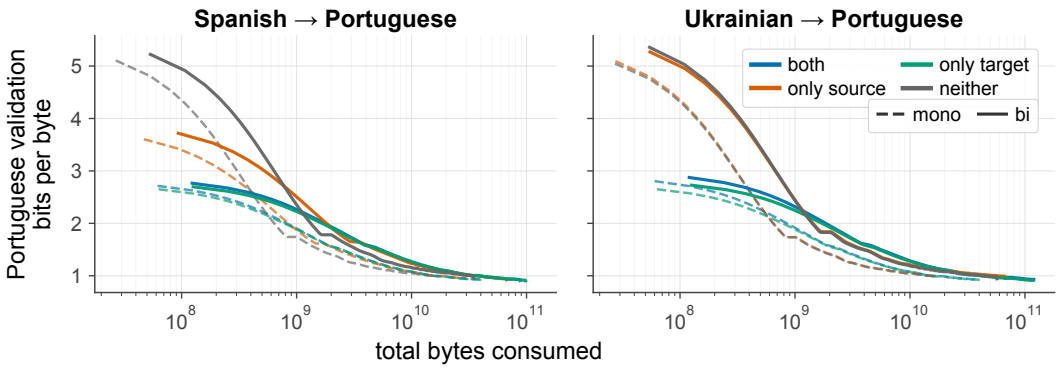

Figure 9: **Learning curves for Portuguese-target transfer.** We compare two source languages for Spanish (left) and Ukrainian (right). Curves show monolingual and 50/50 bilingual training under the four tokenizer conditions. Spanish–Portuguese shows consistently small $\mathrm{BTS}^{T}_{s\to t}$ gaps across tokenizer conditions and positive $\mathrm{BTS}^{\text{both}}_{s\to t}$, whereas Ukrainian–Portuguese exhibits a larger negative $\mathrm{BTS}^{\text{both}}_{s\to t}$ and a very low $\mathrm{BTS}^{\text{only source}}_{s\to t}$.

G Variation of $\mathrm{BTS}^{T}_{s\to t}$ versus $\mathrm{TBTS}_{s\to t}$

We replace the mean of $\mathrm{BTS}^{T}_{s\to t}$ by the obtained $\mathrm{TBTS}_{s\to t}$ using the mixed-effects models. In Figure 10, we can still observe similar patterns as in Section 5.1, that high $\mathrm{TBTS}_{s\to t}$ pairs vary little across tokenizer conditions, while low $\mathrm{TBTS}_{s\to t}$ pairs are much more sensitive.

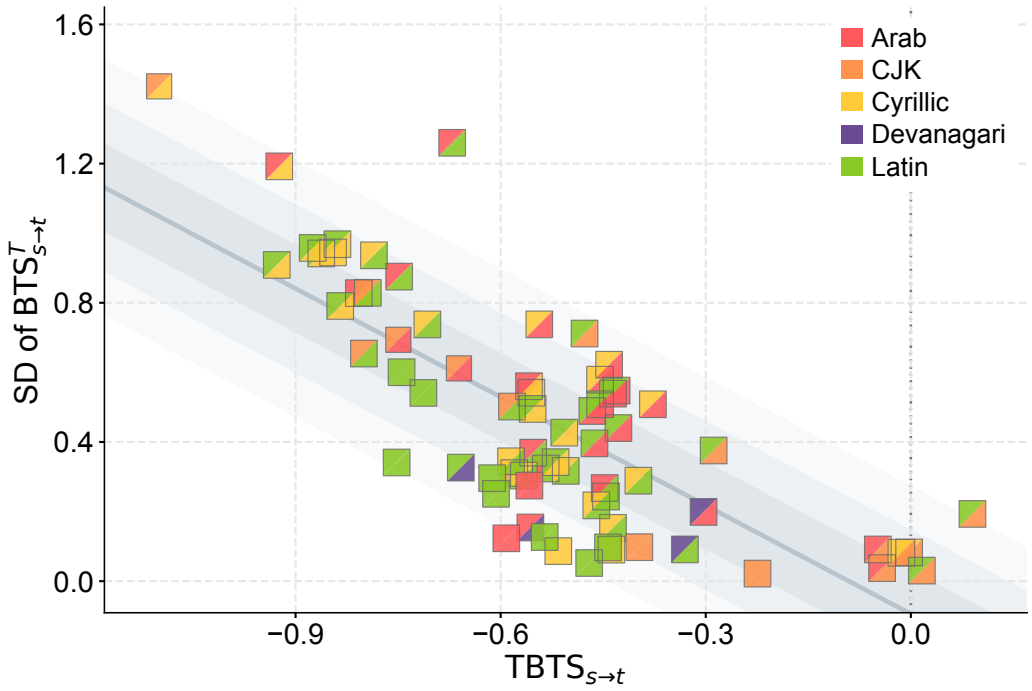

Figure 10: **Sensitivity of $\mathrm{BTS}^{T}_{s\to t}$ estimates as a function of $\mathrm{TBTS}_{s\to t}$.** Each point is one directed language pair. Higher values indicate pairs whose $\mathrm{BTS}^{T}_{s\to t}$ estimate varies more across the different tokenizer settings. Square markers: the upper-left triangle shows the source script and the lower-right triangle shows the target script. Within-script pairs appear as single-color squares.

H Individual Predictor Profiles

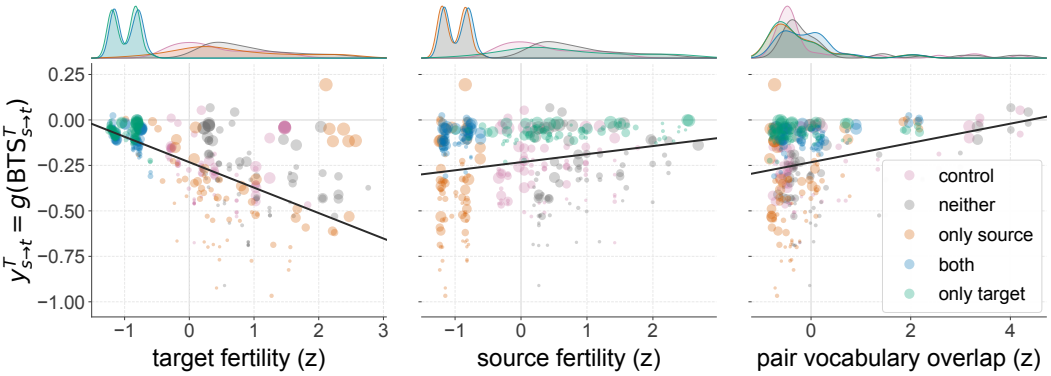

Figure 11: **Another decomposition of BTS$^T_{s\to t}$** Each point is one pair under a certain tokenizer condition. The left three panels show the three individual predictors used in the mixed-effects model, while the heatmap on the right shows TBTS$_{s\to t}$ for each language pair.

Figure 11 visualizes the decomposition. The three left sub-panels project the observations onto each individual predictor. Target fertility separates neither, only source and control (target unseen) on one side, while both and only target (target seen) on the other. Source fertility axis, instead, isolates only the only target condition, which overlapped with both in the target fertility axis. However, vocabulary overlap shows less visible separation among the conditions on its own.

I Case Study of Metrics

We provide another regression using different metrics from that of Section 3.2. This combination of metrics, not symmetric for the source and the target language, provides a slightly different pattern, while still giving a reasonable set of TBTS.

I.1 Metric Choice

Target side, bigram entropy. Section 5 identified the target as the side most associated with BTS$^T_{s\to t}$, and the relevant property is how well the tokenizer segments the target language. We measure this with token bigram entropy rather than fertility or unigram entropy. Poelman et al. (2025) show that token-bigram statistics are an intrinsic, training-free proxy for causal language-modeling difficulty and a gradient proxy for morphological complexity, whereas efficiency-based metrics such as fertility are precisely the factors they identify as confounds when relating tokenization to modeling difficulty. Section 5.1 already showed that scripts are strongly entangled with the tokenizer-associated variation in our data; bigram entropy therefore gives a target-side measure that is annotation-free and robust to those confounds.

Source side, vocabulary utilization. Source coverage shows only a marginal correlation with BTS$^T_{s\to t}$ (Section 5), indicating that the source contributes through coarse representation rather than fine segmentation quality. Using the same metric family on both sides would be redundant and could introduce collinearity: under symmetric coverage (the both and neither conditions), the tokenizer treats the two languages alike. Complementing a target-side segmentation-quality metric with a source-side allocation metric keeps the predictors on distinct axes. We therefore use vocabulary utilization (the fraction of the source language's vocabulary the tokenizer activates) as an allocation measure of how much representational capacity the tokenizer devotes to the source language.

Pairwise, vocabulary overlap. Vocabulary overlap is the distributional similarity of the vocabularies between the two languages under T, capturing their similarity in this tokenizer's representation. It stands for the residual pair-level correlation beyond the per-side properties, consistent with the within- vs. cross-script split in Section 5.1.

I.2 RESULTS

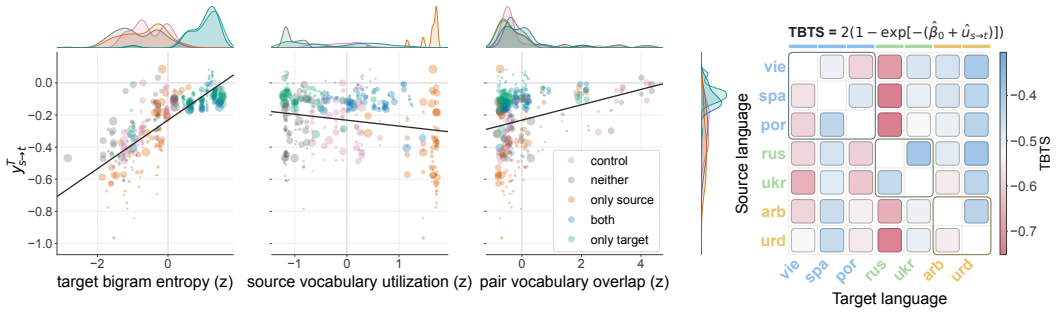

Figure 12: **Another decomposition of BTS$^T_{s\to t}$** Each point is one pair under a certain tokenizer condition. The left three panels show the three individual predictors used in the mixed-effects model, while the heatmap on the right shows TBTS$_{s\to t}$ for each language pair.

We fit eq. (3) jointly with the three predictors above. Figure 12 visualizes the decomposition and the transfer level of pair $u_{s\to t}$ after controlling for tokenizer properties (left). The three left sub-panels project the observations onto each individual predictor. As the response $y^T_{s\to t}$ increases with BTS$^T_{s\to t}$, the sign of each coefficient aligns with the direction of its association. All three fixed effects are significant. Higher target bigram entropy and higher vocabulary overlap are associated with higher BTS$^T_{s\to t}$, while higher source vocabulary utilization is associated with lower BTS$^T_{s\to t}$. The target-side predictor is the most significant by a wide margin ($z = 18.6$, versus $|z| = 4.9$ for source vocabulary utilization and $z = 3.9$ for overlap). This evidence again suggests the target-side correlation with BTS$^T_{s\to t}$ is far stronger, consistent with Section 5, where the target side was the most associated with BTS$^T_{s\to t}$.

The fitted fixed-effect coefficients give the following TBTS$^T_{s\to t}$:

$$\begin{aligned}\text{TBTS}^T_{s\to t} = 2\Big(1 - \exp\Big[-\big(&-0.900\\ &+1.054B_{t,T} - 0.116V_{s,T} + 0.267O_{s,t,T}\\ &+ \hat{u}_{s\to t}\big)\Big]\Big)\end{aligned} \tag{9}$$

Here, $B_{t,T}$ denotes the target-language bigram entropy under tokenizer T, $V_{s,T}$ denotes the vocabulary utilization of tokenizer T on the source language, and $O_{s,t,T}$ denotes the vocabulary overlap for pair $s \to t$ under T.

J PAIR-LEVEL TRANSFER AFTER TOKENIZER ADJUSTMENT

Tokenizer adjustment changes both the values and the ordering of language-pair scores relative to the control tokenizer. Appendix J presents the full matrices and pair-level examples.

We show BTS$^T_{s\to t}$ estimates across the seven languages under the control condition in the left matrix of Figure 13, and the estimated TBTS$_{s\to t}$ on the right for comparison. The two matrices differ in both the values and the ordering of pairs. Given that the regression predicts BTS$^T_{s\to t}$ under held-out tokenizers, we read these differences as tokenizer artifacts in the single-tokenizer estimates rather than estimation noise. We therefore recommend recomputing the estimates of Longpre et al. (2026) with this method, to obtain a resource less tied to one tokenizer.

The absolute level of TBTS$_{s\to t}$ is set using the reference intercept $\hat{\beta}_0^{\text{marg}} + \hat{u}_{s\to t}$ after marginalizing out tokenizer effects, so it is an arbitrary baseline; only the ordering and relative differences across pairs are meaningful. Our result exhibits some relatively high TBTS$_{s\to t}$ for pairs such as Ukrainian $\to$ Russian (-0.471) and Spanish $\to$ Portuguese (-0.438), while TBTS$_{s\to t}$ from other languages to

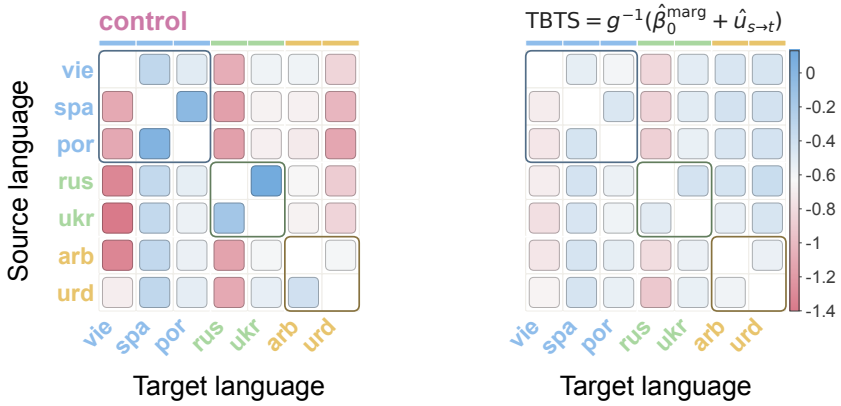

Figure 13: **Heatmaps of $\text{BTS}_{s\to t}^{\text{control}}$ and $\text{TBTS}_{s\to t}$.**

both targets above are all less than -0.5. However, we also observe counterexamples. For instance, Arabic $\to$ Urdu (-0.558) is lower than transfers from all other 5 sources into Urdu (all of which have $\text{TBTS}_{s\to t} > -0.450$), despite them sharing the Arabic script. This suggests that script overlap alone is insufficient to predict better transfer. Unlike the high-transfer cases above, where the languages share both script and family, Arabic and Urdu belong to the Semitic and Indo-Aryan language families, respectively. This may introduce noise that outweighs the benefit of script similarity. In previous settings where tokenizer artifacts were not controlled for (Longpre et al., 2026), such negative effects could have been obscured when either or both languages are tokenized poorly.

K Ablation study of tokenizer conditions.

We conduct another ablation study by removing data from a certain tokenizer condition and show the results in Table 7. While our regression outperforms the baseline under each ablation setting, including all five conditions in the fitting is suboptimal for predicting the $\text{BTS}_{s\to t}^{T}$ under the multilingual tokenizer. We hypothesize that, because the multilingual tokenizer is trained on a number of languages, the data from control and neither represent a very different distribution from the results of this tokenizer. The severe performance drop of MAE from 0.092 to 0.185 after removing both also supports this hypothesis, as the multilingual setting is the most similar to both. This result suggests that data from different conditions may contribute different signals, and the matched tokenizer conditions matter. Therefore, when no prior knowledge of the target tokenizer is available, incorporating data from all conditions is a conservative but not optimal default to ensure generalizability.

Table 7: **Condition ablation.** Best value of each column is in **bold**.

Training conditions	TBTS prediction				Pair-mean baseline			
	MAE ↓	RMSE ↓	Pearson r ↑	Kendall's τ ↑	MAE ↓	RMSE ↓	Pearson r ↑	Kendall's τ ↑
All conditions	0.092	0.105	0.817	0.561	0.315	0.400	0.436	0.222
¬control	0.075	0.090	0.853	0.614	0.296	0.375	0.467	0.250
¬neither	**0.062**	**0.075**	**0.869**	**0.640**	0.270	0.349	**0.485**	**0.271**
¬only source	0.069	0.078	0.845	0.582	**0.196**	**0.248**	0.380	0.141
¬both	0.185	0.244	0.719	0.494	0.412	0.522	0.385	0.206
¬only target	0.118	0.143	0.771	0.549	0.430	0.534	0.427	0.217

L LIMITATIONS

Experimental Scale. Our experiments are restricted to a limited scale. (1) For model training, we only use ∼500M models and a 64K vocabulary size, constraining our interpretation of BTS to relatively smaller models. Our training budget may also be insufficient, as the same amount of data in terms of bytes is equivalent to a smaller number of tokens for languages with *relatively low* fertility on a specific tokenizer. (2) For tokenizer variation, to maximize this axis, the five conditions we choose mainly stand for extreme cases, including zero-shot (neither, only source, only target) and monolingual (control) ones. They do not represent the widely used multilingual tokenizers, which generate fairer and overall better tokenizations on multilingual data. (3) Our language set consists of 70 directional language pairs covering 14 languages across 7 scripts, but only 7 languages across 3 scripts form a complete matrix. The limited number of language pairs restricts our ability to analyze certain subsets or buckets. For instance, although we identify language family in Appendix J as a potentially important factor, the available pairwise data are not balanced enough to enable a meaningful analysis of this dimension.

Limited Evaluation set. We only test our fitted regression on one multilingual tokenizer. While the held-one-condition-out evaluation in Section 6.2 also shows the advantage of using the regression over baseline, we should have more different types of tokenizers from a wide range of training data and algorithms to strengthen the conclusions and interpretations drawn from our ablation study.

Downstream Utility of TBTS$_{s\to t}$ and TBTS$^{T}_{s\to t}$. Though we propose a TBTS$_{s\to t}$ that controls for tokenizers, as well as TBTS$^{T}_{s\to t}$, which extend transfer scores to any measurable tokenizers, their downstream utility has not yet been tested. Currently, no evidence shows that our TBTS$_{s\to t}$ and TBTS$^{T}_{s\to t}$ outperform BTS$_{s\to t}$ in downstream tasks, which is used as an adjustment term in multilingual scaling laws.