Evaluating Multilingual Tokenization under Worst-N Parity-Aware BPE

Vani Kanjirangat[†*], **David Kletz**[†*], **Tanja Samardzic**[†], **Ljiljana Dolamić**[‡], **Fabio Rinaldi**[†]
[†] SUPSI, IDSIA, Switzerland
[‡] armasuisse, Science & Technology, Switzerland
{vani.kanjirangat, david.kletz, tanja.samardzic, fabio.rinaldi}@supsi.ch
ljiljana.dolamic@armasuisse.ch

Abstract

Improving the fairness of a language model is a goal that applies at every level of the model. In this paper, we evaluate a method targeting a foundational level: tokenization. We present a multilingual evaluation of parity-aware tokenization under worst-N optimization, extending PA-BPE to jointly optimize over the N worst-compressed languages. We evaluate this formulation for $N > 1$ across vocabulary sizes of 16K and 32K on the languages from the flores+ benchmark, using metrics that capture both efficiency and structural alignment. Our results reveal that the effects of increasing N are inconsistent across metrics and do not lead to major gains. Efficiency-oriented and boundary-level metrics show a modest tendency to improve at higher values of N, while structural alignment metrics (such as AST alignment and boundary crossing) exhibit no clear pattern, suggesting that compression fairness and linguistic structure are mainly orthogonal objectives. Script-level analysis further reveals uneven effects across writing systems, with several non-Latin scripts showing greater sensitivity to increasing N.

1 Introduction

Tokenization is a foundational step in the training and deployment of large language models (LLMs), responsible for mapping raw text into discrete token sequences that the model processes. While tokenizer training methods are effective for high-resource languages, they tend to favour languages that dominate training data. Nevertheless, tokenization on low-resource languages tends to be of lower quality, a disparity that has consequences at multiple levels. On the one hand, a tokenizer that fragments morphemes, splits identifiers, or poorly covers a script's character inventory undermines the structural fidelity of the representation before learning even begins. As such, this disparity compounds across model training, inference cost, and task performance (Rust et al., 2021; Petrov et al., 2023; Kanjirangat et al., 2025). On the other hand, tokenization efficiency directly governs the number of tokens a model must process per unit of text. Therefore, services billed per token impose higher effective costs on low-resource languages (Petrov et al., 2023; Ahia et al., 2023).

Several approaches have been proposed to address this imbalance. In this paper, we focus on the Parity-Aware BPE algorithm (PA-BPE) (Foroutan et al., 2025). It builds on the popular Byte Pair Encoding (BPE) approach (Sennrich et al., 2016), which constructs a vocabulary by iteratively merging the most frequent byte pairs in a training corpus. Noting that high-resource languages are favoured, Foroutan et al. (2025) replaces the global frequency objective with a max–min criterion: at each merge step, the algorithm selects the merge that most improves the language currently suffering the worst compression rate. This formulation has been shown to improve coverage for low-resource languages without catastrophic loss of aggregate efficiency. Nevertheless, by targeting only the single worst-compressed language at each step, PA-BPE introduces several limitations. First, multiple languages may exhibit similarly low compression rates. Second, restricting merges to statistics drawn from a single language weakens the frequency signal that BPE relies on for stable pair selection, potentially introducing noise into the merge sequence. Building on these observations, we propose a generalisation: jointly optimising the N worst-compressed languages simultaneously.

We present a systematic multilingual evaluation of **worst-N parity-aware tokenization**, extending the PA-BPE baseline to $N \in \{1, 2, 3, 10, 20\}$ across vocabulary sizes of 16K and 32K on the flores+ benchmark (Team et al., 2022). Using a diverse suite of intrinsic metrics spanning compression efficiency, distributional balance, and struc-

[*] Equal contribution

Proceedings of the 1st Workshop on Multilinguality in the Era of Large Language Models (MeLLM 2026), pages 221–228

tural alignment, we study how increasing N affects tokenization quality across languages and writing systems, and whether broader optimisation introduces trade-offs between efficiency and structural fidelity.

Our main contributions are as follows. We propose and evaluate the worst-N generalization of PA-BPE, showing that improvements across metrics are inconsistent and do not lead to major gains. We further demonstrate that the computational cost of increasing N is not justified by the marginal improvements observed, making worst-N optimization with large values of N not recommended.

2 Parity-Aware BPE and Worst-N Objective

We build on Foroutan et al. (2025), which introduces a parity aware objective to optimize the BPE merges. Classical BPE selects merges based on a global frequency objective (Sennrich et al., 2016), which implicitly favors high-resource languages that dominate the training corpus. In contrast, Parity-Aware BPE (PA-BPE) adopts a max–min formulation that prioritizes the language with the poorest compression at each step. Formally, PA-BPE seeks a merge sequence $\mathbf{m}^\star = [m_1, \ldots, m_K]$ that maximizes the minimum per-language compression rate:

$$\mathbf{m}^\star = \max_{\mathbf{m}:\,|\mathbf{m}|=K} \min_{\ell} \mathrm{CR}(\ell;\, \tau_{\mathbf{m}}) \tag{1}$$

where the compression rate of a byte-string b under tokenizer τ is defined as

$$\mathrm{CR}(b;\, \tau) = \frac{|b|_u}{|\tau(b)|} \tag{2}$$

with $|b|_u$ denoting the length of b in normalization units u (e.g., characters, bytes). Higher values indicate stronger compression. At each merge step, PA-BPE identifies the language currently suffering the worst compression rate and selects the merge that most improves it, trading a small amount of aggregate compression for cross-lingual fairness.

The original PA-BPE formulation ($N = 1$) optimizes each merge exclusively for the single worst-compressed language. However, in practice multiple languages may share similarly low compression rates at a given step, making it suboptimal to focus the signal on just one. We generalize PA-BPE to a **worst-N objective** that simultaneously targets the N most poorly compressed languages.

Algorithm 1 Worst-N Parity-Aware BPE

Require: Multilingual corpus $\mathcal{M}$, vocabulary size K, parameter N
Ensure: Merge list $\mathbf{m}$, vocabulary V

Initialize $V \leftarrow$ all unique bytes; $\mathbf{m} \leftarrow [\,]$
for $k = 1$ **to** K **do**
  for each language $\ell \in \mathcal{L}$ **do**
    Compute $\mathrm{CR}(\ell;\, \tau)$ on $\mathcal{D}_\ell$
  end for
  Sort $\mathcal{L}$ ascending by CR $\rightarrow$ $[\ell_1, \ldots, \ell_{|\mathcal{L}|}]$
  $\overline{\mathrm{CR}}_N \leftarrow \frac{1}{N} \sum_{i=1}^{N} \mathrm{CR}(\ell_i;\, \tau)$
  $\mathcal{L}_N \leftarrow \{\ell : \mathrm{CR}(\ell;\, \tau) \le \overline{\mathrm{CR}}_N\}$
  Compute pair frequencies over $\bigcup_{\ell \in \mathcal{L}_N} \mathcal{D}_\ell$
  $m_k \leftarrow$ most frequent pair
  $\mathbf{m} \leftarrow \mathbf{m} \oplus [m_k]$; $V \leftarrow V \cup \{m_k\}$
  Apply merge m_k to all sequences
end for
return $\mathbf{m}$, V

Concretely, at each merge step we compute the compression rates $\{\mathrm{CR}(\ell;\, \tau)\}_{\ell=1}^{L}$ across all languages, then define the *worst-N average* as the mean compression rate over the N lowest-performing languages:

$$\overline{\mathrm{CR}}_N = \frac{1}{N} \sum_{i=1}^{N} \mathrm{CR}(\ell_i;\, \tau) \tag{3}$$

The selected merge is then the one that maximizes improvement for all languages ℓ with $\mathrm{CR}(\ell;\, \tau) \ge \overline{\mathrm{CR}}_N$, i.e., those performing at or below the worst-N mean. This generalizes the $N = 1$ baseline smoothly: when $N = 1$ the formulation reduces to the original PA-BPE max-min objective; as N grows, the optimization broadens toward a larger (and more diverse) set of under-served languages.

The motivation is twofold. First, restricting optimization to a single language at each step may leave a cluster of similarly disadvantaged languages unaddressed. Second, averaging over N languages provides a smoother, more stable frequency signal for BPE pair selection. As we show in our experiments, however, increasing N also dilutes this signal, weakening the pair-selection stability that BPE relies on and producing diminishing returns beyond small values of N.

Training Procedure and Implementation Details We implement a modified version of the original Parity-Aware BPE algorithm by (Foroutan

et al., 2025), described by the pseudocode in Algorithm 1.

The thresholding mechanism was introduced in the original BPE to limit the growth of the pair candidates to consider. Indeed, the frequency of a symbol pair can only decrease over training: each merge replaces occurrences of the pair with a composite symbol, so a pair whose frequency is already low can never become the best candidate. The original threshold was set to one tenth of the maximum frequency observed for language l. However, this static threshold proves too aggressive late in training when global frequencies dropped considerably. We therefore introduce an adaptive threshold defined at iteration t, as $\tau_l^{(t)} = \left(\max_p f_p^{(l)}\right) \cdot \frac{t}{t+C}$, where $\max_p f_p^{(l)}$ is the current maximum frequency over language l and $C = 10000$. This keeps the threshold close to zero at the beginning of training and lets it converge toward the current maximum frequency by the end of training.

Furthermore, selecting the N least-compressed languages requires a full sort over all L languages. To optimize the code, we reduce the frequency of the sort: the indices of the N worst languages are computed once and then reused for k consecutive iterations, with k fixed to 100 in our experiments.

3 Data Sampling & Statistics

We train our tokenizer using FineWeb data for English and FineWeb-2 (Penedo et al., 2025) for the other languages, and evaluate it on the flores+ dataset (Team et al., 2022) . From the languages available in FineWeb-2, we select those that are also included in flores+, yielding 184 languages.

We further decide to group these languages by language family rather than treating them individually, following the classification from WALS (Dryer and Haspelmath, 2013). We make two exceptions: English and Chinese are each treated as a full family (rather than belonging to Indo-European and Sino-Tibetan respectively). In total, we obtain 40 distinct families, listed in Table 3 in Appendix A along with the number of languages each contains. Note that families vary considerably in size, ranging from a single language to over twenty.

Grouping languages implies that the "worst-language" selection in our algorithm now operates at the level of language families rather than individual languages.

To ensure balanced distributions, we perform sampling at both the family and language levels. The total training corpus size is fixed in advance at 1 GB. Sampling proceeds in iterative rounds: as long as the total size has not been reached, $n = 10$ lines are drawn from each available family, ensuring an equal amount of text from each family. If all the text from a given family has been exhausted, that family is excluded from next rounds. Within a family, we also aim for equitable sampling across languages: an internal selection order is defined randomly, and at each round the next language in that order is sampled.

In total, to reach 1 GB of data, we sampled 120,641 lines. On average, each family contains approximately 21 million pre-tokens, and each pre-token has an average length of 4.5 characters. A per-family breakdown is provided in Appendix A.

4 Evaluation Metrics

Downstream task performance remains the ultimate measure of tokenizer quality but would be computationally prohibitive to conduct through full model training at this scale. By contrast, intrinsic evaluations provide a controlled and interpretable lens for isolating the effects of tokenization design choices independent of model architecture and training (Zouhar et al., 2023; Schmidt et al., 2024; Tsvetkov and Kipnis, 2024).

We thus evaluate tokenizers using intrinsic evaluation from the suite of Meister (2025), considering two broad categories: compression and distributional metrics, and structural alignment metrics.

Compression & Distributional Metrics. We report **Fertility**, the average number of tokens produced per normalization unit (line/bytes/words), and **Compression Rate (CR)**, the ratio of original to tokenized sequence length, both of which capture token efficiency. Following the evaluation suite, we use *lines* as the normalization unit, where each line in flores+ corresponds to a semantically equivalent sentence across languages. This measures tokenizer cost per sample rather than linguistic word structure, making it a proxy for computational cost and context window utilization across languages. **Vocabulary Utilization** measures the fraction of vocabulary entries actively used on a given corpus, while **Average Token Rank** reflects the mean frequency rank of produced tokens – lower values indicate over-reliance on a small set of high-frequency entries. Finally, the **Gini Coefficient** quantifies inequality in the token frequency distribution–a lower value indicates more

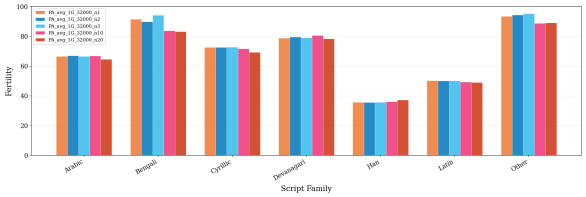

Figure 1: Fertility per script

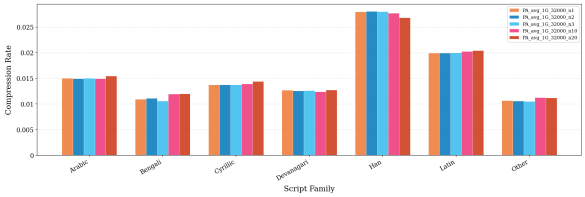

Figure 2: Compression Rate per script

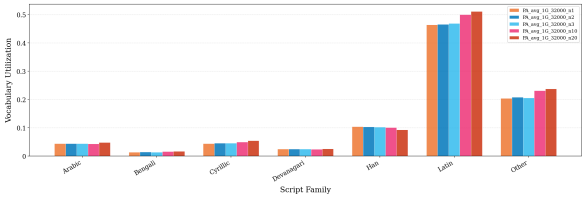

Figure 3: Vocabulary Utilization per script

uniform vocabulary usage, which is particularly desirable in multilingual settings where skewed distributions may signal poor coverage of low-resource languages.

Structural Alignment Metrics. To assess how well token boundaries respect meaningful linguistic and syntactic units, we report five alignment-oriented metrics. **3-Digit Alignment F1** measures whether numeric sequences are tokenized as coherent units rather than fragmented. **AST Alignment** evaluates the correspondence between token boundaries and syntactic constituents in abstract syntax trees, providing a code-aware structural signal. **Identifier Fragmentation** quantifies how often code identifiers such as variable and function names are split across multiple tokens. **Boundary Crossing** captures the rate at which token boundaries violate morphologically or syntactically meaningful unit boundaries. **Character Split Rate** reports the proportion of characters reduced to single-character tokens, indicating insufficient subword coverage.

5 Results

Compression and distribution metrics are given in Table 1, and structural metrics in Table 2. These results reveal that extending parity-aware optimization beyond $N{=}1$ does not yield consistent or monotonic improvements across either compression or structural alignment metrics. While vocabulary utilization increases steadily with N, most other metrics (fertility, Gini coefficient, and the structural alignment measures) exhibit non-monotonic behaviour with no clear trend, and the gains observed even at $N{=}20$ remain modest. This is particularly striking given that $N{=}20$ requires between $\sim$ 14 (with a vocabulary size of 16K) and 70 hours (32K) of training compared to $\sim 2-3$ hours for $N{\leq}2$, making such marginal improvements practically unjustifiable given the computational cost.

We further complement these results with a per-script comparison of Fertility (Figure 1), Compression Rate (Figure 2), and Vocabulary Utilization (Figure 3). These figures reveal that the effects of increasing N are not only non-monotonic, but also inconsistent in direction across scripts. Most strikingly, while scores tend to improve with increasing N for Latin, Arabic, and Cyrillic scripts, they consistently degrade for Han.

Discussion These findings are counterintuitive: broadening the optimization target to include more under-served languages should, in principle, produce a fairer and more balanced tokenizer. We hypothesize that the failure of worst-N to deliver on this promise stems from a fundamental weakening of the frequency signal that BPE relies on for stable merge selection. In the $N{=}1$ case, merge statistics are computed over the single worst-compressed language, producing a sharp, peaked frequency distribution that yields unambiguous, high-confidence merge decisions. As N increases, statistics are pooled across an increasingly diverse set of languages with different scripts, morphologies, and token distributions. This aggregation produces a flatter, more diffuse frequency signal in which no single pair dominates clearly, leading to noisier merge selections and diminishing returns. In the limit, the worst-N average compression rate approaches the global mean, effectively recovering the same implicit bias toward high-resource languages that PA-BPE was designed to correct.

A further issue is that our criterion of targeting all languages at or below the worst-N average may not be effective. The average is sensitive to the specific composition of the worst-N set at each step, which can shift substantially between merge

Table 1: Compression and efficiency metrics for PA worst-N tokenizers [16K,32K] vocabulary size across flores+ languages.

Tokenizer	Voc. Size	**Fertility** ↓	**CR** ↑	**Vocab Util.** ↑	**Avg Token Rank** ↓	**Gini** ↓
PA_1	16K	80.70	0.012	0.767	933.4	**0.265**
PA_2	16K	81.96	0.012	0.768	**884.4**	0.275
PA_3	16K	81.05	0.012	0.785	930.3	0.274
PA_{10}	16K	80.27	0.012	0.800	960.5	0.271
PA_{20}	16K	**78.85**	**0.013**	**0.816**	1047.0	0.270
PA_1	32K	63.636	0.016	0.733	2056.0	0.182
PA_2	32K	63.805	0.016	0.737	2038.3	0.185
PA_3	32K	64.053	0.016	0.736	**2023.1**	0.189
PA_{10}	32K	62.051	0.016	0.789	2259.6	**0.176**
PA_{20}	32K	**61.473**	0.016	**0.807**	2363.9	0.177

Table 2: Structural alignment metrics for PA worst-N tokenizers at [16K,32K] vocabulary size. N=1 achieves the best AST alignment and N=2 the best digit alignment, while structural improvements at higher N are marginal and non-monotonic.

Tokenizer	Voc. Size	**3-Digit F1** ↑	**AST Align.** ↑	**Ident. Frag.** ↓	**Bound. Cross** ↓	**Char Split** ↓
PA_1	16K	0.578	**0.986**	0.783	0.0307	0.2445
PA_2	16K	**0.716**	0.978	0.785	0.0343	0.2689
PA_3	16K	0.638	0.977	0.768	0.0302	0.2523
PA_{10}	16K	0.686	0.981	0.762	0.0312	0.2460
PA_{20}	16K	0.689	0.981	**0.709**	**0.0220**	**0.2214**
PA_1	32K	0.617	**0.954**	0.697	0.0464	0.0811
PA_2	32K	0.632	0.937	0.686	0.0469	0.0829
PA_3	32K	0.659	0.937	0.681	0.0486	0.0877
PA_{10}	32K	0.683	0.928	0.672	0.0373	0.0614
PA_{20}	32K	**0.686**	0.928	**0.658**	**0.0298**	**0.0519**

iterations as compression rates evolve. This instability means the set of languages being optimized is inconsistent across steps, preventing the sustained directional improvement that the N=1 formulation achieves by always anchoring to the single most disadvantaged language. Taken together, these results suggest that the worst-N generalization, while theoretically motivated, undermines the very mechanism that makes PA-BPE effective: a concentrated, stable signal that reliably lifts the language in greatest need at each step. Small-N configurations, particularly N=1–2, appear to strike the most effective balance, and future work might explore adaptive or weighted alternatives that broaden coverage without sacrificing signal quality.

6 Conclusion

In this article, we proposed an adaptation of PA-BPE that selects token pairs maximizing frequency over the N worst-performing languages rather than just the worst language. We trained tokenizers with $N \in \{1, 2, 3, 10, 20\}$ and vocabulary sizes (16K and 32K), evaluated via intrinsic metrics spanning compression efficiency, distributional balance, and structural alignment.

Our results reveal three key findings. Worst-N optimization yields non-monotonic improvements: compression gains concentrate at N=20 versus $N \leq 2$, while structural alignment peaks at N=1–2 and degrades at intermediate values. We observe only weak correlation between compression and alignment metrics, supporting the view that these are orthogonal dimensions (Schmidt et al., 2024). This suggests that small-N optimization (N=1–2) offers the best trade-off: minimal computational cost with strong alignment. We advise against the average-based worst-N threshold as a general heuristic, since it conflates two independent objectives.

7 Limitations

Our study is subject to two limitations. First, we evaluated tokenizers with maximum vocabulary sizes of 32K, substantially smaller than contemporary LLM practice (128K–256K tokens). This constrains the generalizability of our findings to production-scale models. Second, our evaluation relies exclusively on intrinsic metrics, while

downstream task performance across diverse downstream tasks would strengthen the validity of our results.

Acknowledgments

The work described in this paper has been supported by the "Balancing Multilingual token vocabulary (BM-Tok)" project, funded by armasuisse S&T, Switzerland (Project CYD-C-2020004). We also thank the Swiss AI Initiative for providing access to their computational infrastructure

References

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923, Singapore. Association for Computational Linguistics.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online (v2020.4)*. Zenodo.

Negar Foroutan, Clara Meister, Debjit Paul, Joel Niklaus, Sina Ahmadi, Antoine Bosselut, and Rico Sennrich. 2025. Parity-aware byte-pair encoding: Improving cross-lingual fairness in tokenization. *Preprint*, arXiv:2508.04796.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Vani Kanjirangat, Tanja Samardzic, Ljiljana Dolamic, and Fabio Rinaldi. 2025. Tokenization and representation biases in multilingual models on dialectal NLP tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 23992–24010, Suzhou, China. Association for Computational Linguistics.

Clara Meister. 2025. Tokeval: A tokenizer analysis suite.

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

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, volume 36, pages 36963–36990. Curran Associates, Inc.

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

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Alexander Tsvetkov and Alon Kipnis. 2024. Information parity: Measuring and predicting the multilingual capabilities of language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7971–7989, Miami, Florida, USA. Association for Computational Linguistics.

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

A Training data Statistics

Statistics about language family distribution in our training corpus are available on Table 3

Table 3: Language family distribution with sampled line counts, language counts, pre-token counts, and mean sequence length in characters

File	# Sampled lines	# Langs	# Pre-tokens ($\times 10^4$)	Mean length
Afro-Asiatic	3070	7	233	5.09
Altaic	3070	4	173	6.11
Asmat-Kamrau Bay	3070	2	227	3.12
Austro-Asiatic	3070	2	302	3.29
Austronesian	3062	19	228	4.56
Cariban	3060	1	143	4.23
Central Sudanic	3070	1	122	5.97
Chapacura-Wanham	2258	1	77	5.34
Chinese	3070	1	59	6.7
Choco	3070	1	157	5
Dravidian	3060	1	525	2.03
Eastern Sudanic	3070	2	208	4.9
Fur	3070	1	194	4.7
Hatim-Mansim	3070	1	190	4.58
Hokan	3070	2	279	2.84
Indo-European	3060	24	277	3.68
Japanese	3070	1	54	7.83
Korean	3070	1	147	3.48
Maban	3070	1	113	5.72
Mande	3070	3	155	4.37
Mayan	2910	1	251	4.3
Moraori	3070	1	248	4.16
Mosetenan	2266	1	278	3.71
Na-Dene	3070	2	175	5.13
Niger-Congo	3070	19	294	3.89
North Halmaheran	2752	1	175	5.06
Northwest Caucasian	3070	1	186	4.59
Oregon Coast	3070	1	146	5.14
Pama-Nyungan	3061	2	195	6.1
Salishan	3070	1	240	4.51
Sino-Tibetan	3058	8	364	2.96
Tai-Kadai	3070	3	208	4.21
Tor-Kwerba	3070	1	244	2.14
Trans-New Guinea	3070	4	359	3.06
Tucanoan	3070	1	391	2.24
Uralic	3070	2	174	6.37
Zamucoan	3070	1	170	5.87
Zaparoan	3070	1	193	4.97
eng	3070	1	211	4.97
other	3064	60	211	4.74