
Non-English Speakers Pay More for Less: Tokenizer Design Must Be a First-Order Fairness Concern

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 We argue that tokenizer design is a first-order concern for equitable multilingual
2 AI, with direct consequences for cost, performance, and access. Large Language
3 Models are often celebrated as democratizing technologies, yet their fundamen-
4 tal infrastructure embeds systematic bias against most of the world’s languages.
5 This position paper argues that tokenization, defined as the process of converting
6 text into model-readable units, creates profound and largely invisible inequalities.
7 Drawing on the Sapir-Whorf hypothesis, we introduce the concept of *cognitive*
8 *friction* to explain how misaligned tokenization degrades not merely efficiency
9 but the fundamental quality of language understanding. We trace the cascading
10 consequences of this inequity: a language tax that makes LLM-based services
11 used in non-English languages systematically more expensive, degraded reasoning
12 performance as sequence lengths inflate, and disproportionate environmental costs
13 carried by users of inefficiently tokenized languages. Current benchmarks and
14 “multilingual” marketing claims obscure these disparities, creating an illusion of
15 parity that does not exist. We call for transparency in tokenizer efficiency reporting,
16 research into language-adaptive and equitable tokenization strategies, investment
17 in language-specific foundation models, and policy frameworks that treat linguistic
18 equity as a first-order concern. Tokenization is an active design decision that
19 determines who benefits from AI and who bears hidden costs.

20 1 Introduction

21 Large Language Models (LLMs) are increasingly framed as democratizing technologies, yet beneath
22 this promise lies a structural inequity: tokenization. Of the world’s approximately 7,000 languages,
23 only 7 (0.28%) qualify as “winners” with abundant digital resources, which are integrated into
24 the training of LLMs, while 88% are “left behind” due to lack of resources [Joshi et al., 2021].
25 Tokenization is both a symptom and amplifier of this disparity.

26 Tokenization, the process of converting text into discrete units for model processing, is often treated
27 as a mere preprocessing step. This neglect is a serious oversight. The design of a tokenizer determines
28 how efficiently a language can be represented, and current tokenizers are far from language-neutral.
29 A tokenizer trained predominantly on English efficiently encodes common words as single tokens,
30 but when applied to German, Arabic, or Swahili, it may require two, three, or even ten times as many
31 tokens for equivalent content. This mirrors what we know from human language comprehension: the
32 structure and vocabulary of a language influence how its speakers perceive and think about the world,
33 known as the (weak-form) Sapir-Whorf hypothesis Whorf [1956]. In the case of LLMs, they inherit
34 an English-centric orientation through tokenization.

35 This creates cascading consequences: (1) users pay more for API access [Ahia et al., 2023, Tekle-
36 haymanot and Nejdli, 2025], (2) higher energy consumption and carbon footprint [Solatorio et al.,

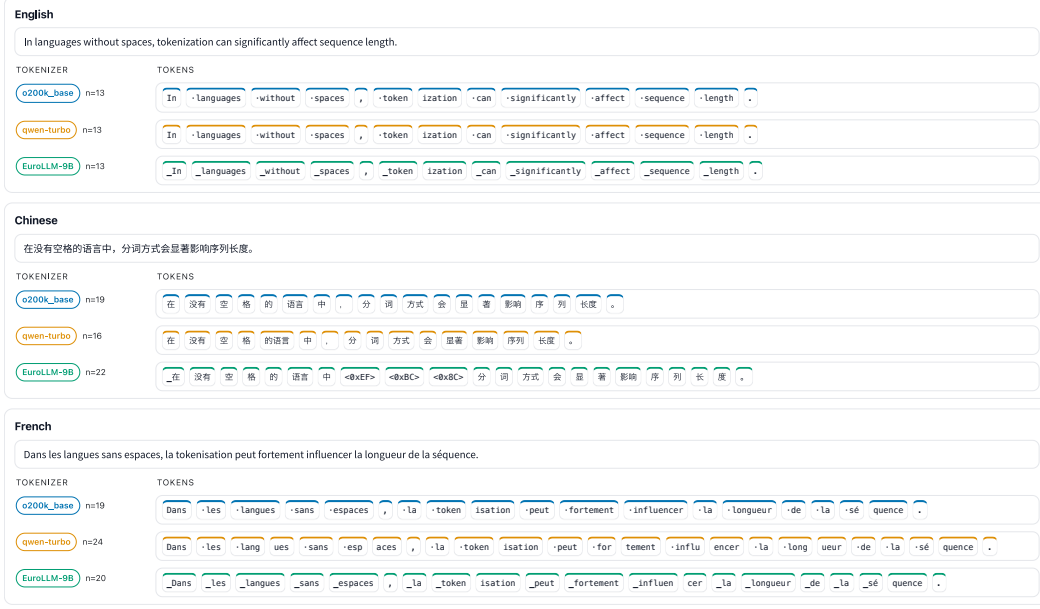


Figure 1: Tokenization efficiency varies across languages and tokenizers. The same semantic content requires different numbers of tokens depending on the tokenizer-language pairing, illustrating the systematic bias embedded in current tokenization approaches.

2024, Ali et al., 2024], and (3) degraded reasoning as sequences grow longer, straining attention mechanisms. Comparing three tokenizers (OpenAI’s GPT o200k, English-adjacent; Alibaba’s Qwen tokenizer, Chinese-English-adjacent; and EuroLLM’s tokenizer, strong multilingual coverage claim) reveals stark disparities: the same semantic content requires vastly different computational resources depending on the tokenizer-language pairing. We note that these three (common) tokenizers have been chosen as examples demonstrating the disparities within and between tokenizers. Following the evidence from ongoing research, which we present below, we make the following call to the community:

Position: Tokenizer design is a first-order concern for fair and efficient multilingual AI. The field must move beyond treating tokenization as an auxiliary problem and instead recognize it as a critical bottleneck that shapes many downstream aspects, including cost, environmental impact, and performance across languages. We call for (1) awareness and transparent reporting of tokenizer efficiency across languages, (2) research into mitigation strategies and language-adaptive tokenization, and (3) investment in language-specific or genuinely multilingual models that do not systematically disadvantage the majority of the world’s speakers. In this paper, we make four main contributions:

1. **Empirical analysis of tokenization disparities.** We compare three prominent tokenizers (GPT o200k, Qwen, EuroLLM) across 20+ languages, documenting efficiency disparities ranging from $1.3\times$ to over $15\times$ relative to English. Non-Latin scripts and morphologically complex languages suffer disproportionately.
2. **Theoretical framework: Cognitive friction.** Drawing on the weak-form of the Sapir-Whorf hypothesis, we introduce *cognitive friction* to explain how misaligned tokenization degrades efficiency and the fundamental quality of language understanding. English-centric tokenization, or more precisely an English-centric allocation of the tokenization budget, imposes English-like semantic structures on all languages, creating systematic disadvantages beyond cost.
3. **Cascading consequences analysis.** We trace the cascading inequities of “double jeopardy” [Solatorio et al., 2024]: speakers of token-inefficient languages face both elevated costs *and* degraded performance simultaneously.
4. **Community agenda for linguistic equity.** We call for transparency in tokenizer efficiency reporting, cost-aware evaluation frameworks, and investment in language-adaptive tokeniza-

tion strategies. We provide actionable recommendations to address tokenization inequity as a first-order concern.

Scope and epistemic status. Throughout this paper, we distinguish three types of claims: (i) claims supported by our own original empirical analysis (the three-tokenizer comparison across 20+ languages on FLORES-200, TED, and UN corpora; Figures 2–3, Appendix Figure 4, and Table 1), (ii) findings synthesized from existing literature, which we attribute to the respective authors, and (iii) broader interpretive hypotheses—notably the *cognitive friction* framework (Section 4.1)—that we advance for community discussion. While prior work has documented individual aspects of the tokenization problem in isolation (cost disparities: Ahia et al. 2023; efficiency metrics: Petrov et al. 2023; fairness-aware tokenization: Foroutan et al. 2025), our contribution lies in providing a unified framework connecting tokenization to cost, performance, reasoning quality, environmental impact, and market dynamics through the lens of cascading consequences.

2 Why Tokenization Matters

Tokenization converts text into discrete tokens (integer identifiers) for model processing. Modern LLMs predominantly use Byte Pair Encoding (BPE) [Sennrich et al., 2016], which iteratively merges frequent character pairs until reaching a target vocabulary size. Crucially, while any string can be tokenized in exponentially many ways ($> 10^{267}$ for a single paragraph under Llama2; Geh et al. 2024), LLMs rely on a single canonical encoding that reflects the statistical biases of the training corpus. Languages underrepresented in that corpus receive suboptimal encodings by construction: “tokenization” may become one token while the German “Tokenisierung” requires three or four (see Figure 1).

Language	Script	Multiplier	Reference
Spanish	Latin	$1.3\times$	Solatorio et al. [2024]
French	Latin	$1.3\times$	Solatorio et al. [2024]
Chinese	Han	$1.3\times$	Solatorio et al. [2024]
Chinese	Han	$1.7\times$	Yang [2024]
Kannada	Kannada	$2.19\times$	Kanjirang et al. [2025]
Korean	Hangul	$2.6\times$	Seo et al. [2025]
Telugu	Telugu	up to $5\times$	Ahia et al. [2023]
Georgian	Georgian	up to $5\times$	Ahia et al. [2023]
Indic languages	Various	$5\times$	Ahia et al. [2024]
Bengali	Bengali	$\sim 6\times$	Rahman et al. [2024]
Myanmar	Myanmar	$\sim 7\times$	Teklehaymanot and Nejdli [2025]
Burmese	Myanmar	up to $10\times$	Yang [2024]
Amharic	Ethiopic	up to $10\times$	Yang [2024]
Dzongkha	Tibetan	up to $14\times$	Solatorio et al. [2024]
Santali	Ol Chiki	up to $14\times$	Solatorio et al. [2024]
Shan	Tai Tham	up to $15\times$	Petrov et al. [2023]

Table 1: Examples of token count multipliers relative to English across languages and scripts. Most ratios are reported for English-optimized tokenizers (especially GPT tokenizer).

2.1 Theoretical Foundations

Recent theoretical work provides formal justification for why tokenization is essential for transformer performance. Rajaraman et al. [2025] demonstrate that without tokenization, transformers fail to learn correct distributions from k -th order Markov processes, instead predicting characters according to a unigram model with high cross-entropy loss. With appropriate tokenization, transformers achieve near-optimal performance. This theoretical grounding confirms that tokenization is not merely a preprocessing convenience but a fundamental enabler of transformer learning capacity (see Appendix B for detailed theoretical results).

96 2.2 Vocabulary Allocation, Encoding, and the Zero-Sum Constraint

97 Tokenization operates under a fundamental constraint: vocabulary size is fixed. This creates a zero-
98 sum allocation problem where vocabulary slots allocated to one language are unavailable to others.
99 Limisiewicz et al. [2023] formalize this through the concept of *vocabulary allocation*, referring
100 to the portion of the multilingual vocabulary devoted to meaningful units of each language. They
101 demonstrate that higher allocation strongly correlates with downstream task performance (Spearman
102 $\rho > 0.65$ for word-level tasks). Vocabulary slots allocated to English morphemes are unavailable
103 for German compounds or Chinese characters, meaning design choices about which languages
104 to prioritize have direct consequences for all others. This zero-sum nature explains why broad
105 multilingual coverage often results in universal mediocrity: Attempting to serve many languages
106 within a fixed vocabulary size necessarily requires tradeoffs (see Appendix C for detailed analysis).

107 Beyond vocabulary allocation, structural encoding constraints create additional inequities. Unicode
108 encoding disparities mean that Latin scripts require approximately 1 byte per character, Cyrillic
109 requires 2, and Indic scripts require up to 4 bytes per character under UTF-8 [Ahia et al., 2024].
110 Even with fairness interventions like α -sampling during BPE training, persistent $5\times$ token count
111 disparities remain between Indic and Latin languages, demonstrating that data rebalancing alone
112 cannot overcome structural bias. Byte-level tokenizers, often presented as a fair alternative, still show
113 bias due to these Unicode encoding efficiency disparities [Remy et al., 2024], meaning no current
114 approach achieves true language neutrality (see Appendix C).

115 3 Tokenizer Effects on Language-(In)Efficiency

116 A tokenizer learns which character sequences to merge based on their frequency in the training corpus.
117 If that corpus is predominantly English, English words and subword patterns will dominate the
118 vocabulary. Common English morphemes like “-tion”, “-ing”, and “pre-” will be assigned dedicated
119 tokens, while equivalent patterns in other languages will be fragmented into smaller pieces.

120 Languages well-represented in training corpora receive efficient encodings, while underrepresented
121 languages are forced into inefficient character-level or byte-level representations [Rust et al., 2021,
122 Ahia et al., 2023, Teklehaymanot and Nejd, 2025]. The problem compounds for languages with
123 non-Latin scripts, complex morphology, or agglutinative structure. Kanjirangat et al. [2025] reveal a
124 particularly troubling failure mode: byte-level tokenizers produce misaligned tokens for non-Latin
125 scripts, interpreting Arabic and Hindi characters as Latin-1 byte sequences, fundamentally distorting
126 input representation. Studies document clear patterns: the more diverse a language is compared to
127 English, the higher the tokenization costs (see Table 1); see also Section 3.2 below.

128 3.1 The Language Tax: Pricing Inequity and Infrastructure Bias

129 Commercial LLM APIs charge per token, both for input and output. If a French user requires 1.3
130 times as many tokens as an English user to express the same query and receive an equivalent response,
131 they pay 1.3 times as much. This pricing asymmetry is invisible to users but represents a systematic
132 tax on non-English language use. Tokenization premiums range from $1.3\times$ for languages like Spanish
133 and French to over $15\times$ for languages like Shan (see Table 1 and Table 2 in Appendix J). This
134 inequity is structural, embedded in infrastructure rather than explicit policy. Teklehaymanot and
135 Nejd [2025] term this “infrastructure bias” to emphasize that these disparities arise from foundational
136 design choices in AI systems rather than incidental implementation details. The problem extends
137 beyond passive inequity to active incentive misalignment: pricing models can create incentives where
138 providers extract unfair premiums through tokenization manipulation (see Appendix D for detailed
139 analysis).

140 These cost disparities cascade into broader downstream effects: context window limitations prevent
141 in-context learning for token-inefficient languages (languages with relative token count (RTC) $>$
142 4.0 receive only one-quarter of the usable context window; Teklehaymanot and Nejd 2025), and
143 energy costs scale directly with token count, creating environmental justice concerns. Solatorio et al.
144 [2024] call this “double jeopardy”: speakers face both, elevated costs *and* degraded performance
145 simultaneously (see Appendix E for detailed analysis).

3.2 Empirical Evidence: Comparing Tokenizer Effects across Languages

To support the discussion, we present our own original empirical analysis comparing three prominent tokenizers representing different optimization strategies: o200k, Qwen, and EuroLLM. Using parallel texts from FLORES-200, TED, and UN corpora, we compute fertility rates (tokens per word/character) following the methodology of Rust et al. [2021], Ahia et al. [2023]. All results in Figures 2–3 (and Figure 4 in Appendix A) and the data in Table 1 are from our original analysis across 20+ languages.

Our comparison reveals three distinct patterns: The o200k tokenizer achieves excellent efficiency for English but degrades rapidly for other languages (German requires $\sim 1.5\times$ tokens, Arabic $3\times$ or more). The Qwen tokenizer demonstrates that targeted bilingual optimization works, achieving efficiency for Chinese just like the o200k tokenizer for English, while also remaining competitive for English. The EuroLLM tokenizer presents a cautionary tale: despite multilingual ambitions, it achieves efficient encoding for none of its target languages, illustrating that broad coverage without sufficient vocabulary or training data balance results in universal mediocrity.

Figures 2 and 3 (and additional corpus results in Appendix A) each contain two panels. The left panels show *relative token ratios normalized to English*, computed per parallel sentence pair, which is the meaningful metric for cross-linguistic comparison. The right panels show *absolute tokens per sentence*; because parallel corpora contain translations of varying verbosity, absolute token counts can appear roughly similar across languages, so readers should focus on the left panels for evidence of disparities.

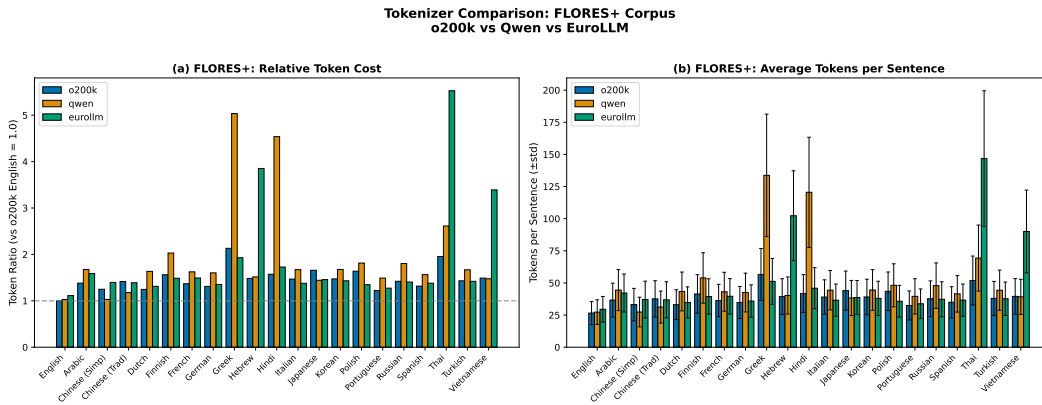


Figure 2: Tokenization efficiency across languages and tokenizers on the FLORES-200 corpus. **Left:** relative token ratio normalized to English ($1.0\times$); this panel controls for cross-linguistic sentence-length variation and shows the meaningful comparison. **Right:** absolute tokens per sentence (note: sentence length varies across languages in parallel corpora; see left panel for the controlled comparison).

Who is most affected? The scale of the problem becomes clear when we consider speaker populations across severity bands. Languages with *mild* penalties ($1.3\text{--}2\times$), such as Spanish, French, and Chinese (under optimized tokenizers like Qwen), affect over one billion speakers and impose moderate cost and context-length penalties. Languages with *moderate* penalties ($2\text{--}5\times$), including Korean, Hindi, Bengali, and Arabic, affect roughly two billion speakers and create significant cost and performance impacts. Languages with *severe* penalties ($5\text{--}15\times$), such as Burmese, Amharic, Khmer, Dzongkha, and many low-resource languages, affect over 200 million speakers and impose near-prohibitive costs with severe context-window reduction.

Importantly, for widely spoken languages such as Chinese, Arabic, and Spanish, the severity depends substantially on *which tokenizer* is used. For example, Chinese ranges from near-parity under Qwen to roughly $3\times$ under o200k (see Figure 2). This observation reinforces rather than undermines our core argument: the disparities are *design choices*, not *technical inevitabilities*. The fact that at least one tokenizer achieves reasonable efficiency for most languages demonstrates that optimization works—but users are typically locked into a single provider’s tokenizer and cannot independently choose the one most favorable for their language.

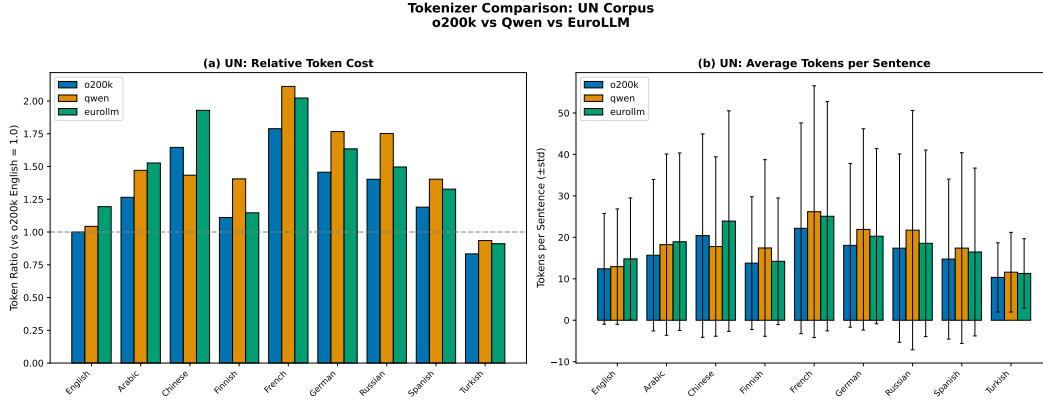


Figure 3: Tokenization efficiency across languages and tokenizers on the UN corpus (formal/legal domain). **Left:** relative token ratio normalized to English (1.0 \times); this panel controls for cross-linguistic sentence-length variation. **Right:** absolute tokens per sentence. Domain specialization amplifies disparities compared to FLORES-200 (Figure 2).

3.3 Non-Latin Scripts and Morphological Complexity Are Penalized

Languages using Arabic, Chinese, Cyrillic, or other non-Latin scripts consistently require more tokens (Table 1), if not specifically trained for (see Qwen and Chinese). Unicode encoding disparities create structural bias: Latin scripts require ~ 1 byte per character, Cyrillic 2, Indic up to 4 bytes. Even with fairness interventions, persistent $5\times$ disparities remain between Indic and Latin languages. Many tokenizers lack complete alphabets for languages spoken by tens of millions (see Appendix C). Agglutinative languages (Turkish, Finnish) and languages with rich inflection (German, Russian) produce longer token sequences because their word forms rarely appear in English-dominated training corpora. BPE’s greedy merging fails to align with morpheme boundaries, introducing semantic ambiguity and destroying morphological structure (see Appendix C).

4 Tokenization’s Impact on Performance: The Sapir-Whorf Connection

The implications of tokenization extend beyond computational efficiency into the nature of meaning itself. The Sapir-Whorf hypothesis, also known as *linguistic relativity*, proposes that the structure of a language influences how its speakers perceive and conceptualize the world [Whorf, 1956, Sapir, 1929]. While the strong deterministic version of this hypothesis remains contested, substantial evidence supports a weaker form: language shapes habitual patterns of thought, categorization, and meaning-making [Boroditsky, 2001, Lucy, 1997].

This theoretical lens casts tokenization in a new light. When a tokenizer fragments text into subword units, it is not performing a neutral encoding; instead, it is imposing a particular structure on meaning. A tokenizer predominantly trained on English implicitly encodes English morphological boundaries, semantic chunks, and conceptual divisions as the “natural” way to segment language. When applied to other languages, this segmentation may cut across meaningful units, separating morphemes that carry unified meaning or grouping characters that should remain distinct.

There is empirical work to support this interpretation. Ray [2025] ran a comprehensive quantitative evaluation of linguistic relativity in AI-generated text, testing ChatGPT-4o mini across 13 typologically diverse languages on culturally salient prompts. The results show distinct conceptual and affective output depending on the language used. Thus, LLMs do not merely translate between languages but impose language-specific conceptual frames on their outputs. Wang et al. [2025a] introduced BICAUSE, a structured bilingual dataset for causal reasoning, and demonstrate that LLMs exhibit *typologically aligned* attention patterns: They find that models internalize language-specific preferences for causal word order and rigidly apply them even to atypical inputs. Chinese reversed causal chains (effect \rightarrow cause) suffer a 15-percentage-point accuracy drop (76.5% vs. 91.2% for canonical order), while English shows only a 2.5-point drop. This indicates greater flexibility of

English causal expression in training corpora. LLMs not only mimic surface linguistic forms but internalize the reasoning biases shaped by language.

Consider an agglutinative language like Turkish or Finnish, where a single word can encode what English expresses in an entire phrase. An English-optimized tokenizer will split such words into fragments that obscure their compositional semantics, forcing the model to reconstruct meaning from pieces that were never meant to stand alone [Asgari et al., 2025]. Similarly, for languages like Chinese or Japanese, where character boundaries carry semantic weight, byte-level fallback tokenization can obscure the very units through which meaning is organized [Wang et al., 2025b].

From a Sapir-Whorfian perspective, tokenization that fragments non-English languages through English-centric segmentation is not merely inefficient—it can be viewed as a form of linguistic imperialism encoded in infrastructure.

The tokenizer becomes a (predominantly) English-shaped bottleneck through which all languages must pass, filtering the model’s worldview through an English-centric lens. Tokenization inequity affects not only cost and speed, but the fundamental quality of the model’s understanding.

4.1 Cognitive Friction and Misaligned Representations

Drawing an *analogy* from the Sapir-Whorf hypothesis to LLM behavior, we introduce the concept of *cognitive friction* to describe the additional processing burden imposed when tokenization boundaries misalign with a language’s natural semantic structure. Just as humans experience cognitive friction when forced to parse information presented in unnatural or fragmented ways [Cooper et al., 2007], language models may need to expend additional computational effort to reconstruct coherent meaning from poorly aligned token sequences. We note that this analogy is intended as an interpretive framework for organizing existing empirical evidence, not as a claim of mechanistic equivalence between human and model cognition.

When a tokenizer respects semantic boundaries, the model receives input that maps naturally onto meaningful units: morphemes, words, and phrases arrive as coherent chunks that align with the structure of concepts. The model can process these units directly, leveraging patterns learned during training. But when tokenization cuts across semantic boundaries, e.g., splitting a Turkish agglutinative verb into arbitrary fragments, or decomposing a Chinese idiom into individual bytes, the model must perform additional work to reassemble meaning. Each unnatural boundary introduces friction: the model must learn to recognize that certain token sequences constitute unified concepts despite being formally separated.

Emerging evidence is consistent with the interpretation that this cognitive friction has measurable consequences. Translation quality degrades for non-Latin languages when inefficiently tokenized [Rust et al., 2021]. Empirical studies document extreme reasoning breakdowns when tokenization boundaries misalign with semantic structure, with error rates reaching 40-98% across models (see Appendix F for detailed linguistic findings, and Table 2 in Appendix J for an overview of studies). Tokenization failures are sensitive to both cross-linguistic and intra-linguistic variation (including dialectal forms and spelling variants), revealing the tensions between meaning preservation and stylistic sensitivity that cannot be simultaneously optimized. Similarly, domain specialization compounds inequity: while English shows minimal fertility degradation, non-English languages require 2-4 times more tokens per word in legal and scientific texts (see Appendix C), creating compounding disadvantages: a base language tax plus an additional domain tax in high-stakes fields where AI assistance is most valuable.

Impact on reasoning and performance. Model performance degrades as sequence length increases [Liu et al., 2024]. When misaligned tokenization inflates sequence length, it may impair reasoning—not because the task is harder, but because the representation is less efficient. While sequence length effects and tokenization quality effects are difficult to fully disentangle, existing evidence is consistent with the interpretation that both contribute to performance degradation. Controlled experiments confirm that tokenizer choice alone accounts for up to 9 percentage point performance gaps on reasoning benchmarks, with systematic error patterns revealing predictable failures [Ali et al., 2024]. Tokenization-induced breakdowns affect even the most advanced models (error rates 40–98% when boundaries misalign; Wang et al. 2025b), and effects are stronger in smaller models, suggesting biases may become more pronounced as the field trends toward compact models (see Appendix G).

5 Alternative Views

5.1 Differences Reflect Inherent Linguistic Diversity

One might argue that tokenization disparities reflect inherent linguistic complexity rather than design bias. Agglutinative languages encode entire phrases in single words; morphologically rich languages exhibit systematic fertility variation across cases [Turuta and Maksymenko, 2025]; Unicode encoding disparities create constraints (Latin ~ 1 byte, Cyrillic ~ 2 , Indic up to 4 bytes; Ahia et al. 2024).

Our reply: While structural properties create constraints, empirical evidence demonstrates dramatic improvements through targeted optimization. The Qwen tokenizer achieves near-parity for Chinese [Yang, 2024]. Rana et al. [2025] achieve $6.4\times$ improvement for Oriya; Foroutan et al. [2025] show Parity-Aware BPE reduces inequality by 83% while maintaining competitive compression. These results demonstrate that efficiency can be substantially improved through design choices rather than being purely determined by inherent properties.

5.2 Compression Argument: Diversity Limits Efficiency

A related argument asserts that tokenization is fundamentally a compression problem: diverse languages in a single vocabulary reduce efficiency for any single language. Information theory implies tradeoffs for a fixed vocabulary size; as evidenced by EuroLLM’s universal mediocrity.

Our reply: Current tokenizers operate far from theoretical limits. EuroLLM’s relatively lower efficiency may plausibly reflect factors such as insufficient vocabulary size or imbalanced training data rather than fundamental compression limits, though without access to its training details this remains a hypothesis [Petrov et al., 2023]. Limisiewicz et al. [2023] show vocabulary allocation strongly predicts performance ($\rho > 0.65$). Foroutan et al. [2025] prove equitable compression is achievable: Parity-Aware BPE achieves 83% inequality reduction with competitive compression. Petrov et al. [2023] show diminishing returns: allocating only one-third of vocabulary to English increases English sequence length by just 10%, suggesting substantial reallocation room.

A related concern is that fairer tokenizers would trade inference efficiency gains against training efficiency losses, since a larger, more balanced vocabulary increases the embedding layer size. This tradeoff is real but likely favors fairer tokenizers: in current deployment cycles, inference tokens vastly outnumber training tokens, so inference-side savings from a better-balanced vocabulary would dominate. Moreover, the current status quo is not necessarily training-efficient either: Ali et al. [2024] found that using English-centric tokenizers can increase computational training costs by up to 68% when multilingual data is included, because misaligned tokenization inflates sequence lengths during training as well. We acknowledge that the optimal balance between vocabulary size, training efficiency, and inference efficiency across languages remains an open research question.

One might further argue that market forces will self-correct these inefficiencies as non-English markets grow. We see three limitations to this reasoning. First, the market mechanism requires *information*: users must be able to observe and compare tokenizer efficiency across providers, yet as we document in Section 6, tokenizer efficiency statistics are rarely reported, creating information asymmetry that our transparency recommendations aim to resolve. Second, *empirically*, the market has had several years to self-correct, yet the disparities we document persist across tokenizer generations (e.g., from GPT-3-era tokenizers to o200k). Third, market forces optimize for the largest revenue-generating segments; the long tail of languages spoken by hundreds of millions—Bengali, Burmese, Amharic, and many others—represents smaller commercial markets despite large speaker counts. The populations most disadvantaged by tokenization are precisely those least likely to be served by market incentives alone.

6 Implications for Research and Deployment

The tokenization inequities we have documented carry significant implications for how the research community evaluates multilingual models and how organizations deploy AI systems globally. Current practices systematically obscure these disparities, leading to overconfident claims and underinformed decisions. Standard multilingual benchmarks typically evaluate models on accuracy, fluency, or task completion without accounting for the computational cost of achieving that performance, creating an

illusion of parity when substantial inequities exist (see Appendix H for detailed analysis of evaluation gaps and recommendations).

6.1 “Multilingual” Claims, Economic Barriers, and Practical Implications

The label “multilingual” has become a marketing term as much as a technical descriptor. Models advertised as supporting 100+ languages may technically process those languages while offering vastly degraded efficiency and capability for most of them. The distinction between coverage (can process) and quality (processes efficiently and effectively) is crucial but rarely surfaced in model documentation. Even within a “supported” language, variation matters: Wegmann et al. [2025] show that English dialects are processed differently by tokenizers trained on Standard American English. We hypothesize that an “English + 1” tokenization strategy—optimizing for English plus *one* target language—might be a practical compromise; this may explain Qwen’s strong performance for Chinese, though without access to Alibaba’s training details, this interpretation remains speculative. Model cards rarely include tokenizer efficiency statistics across languages, leaving users unable to make informed decisions (see Appendix D).

Tokenization inequity also creates material barriers to AI adoption. A Thai startup faces higher API costs than an American competitor for the identical product; a school district in Egypt must accept either higher costs or reduced functionality compared to an American counterpart. Developers rationally focus on English-language applications where costs are lowest, perpetuating global inequalities in technology access.

For *researchers*, these findings suggest several imperatives: (1) report tokenizer fertility rates across target languages as standard practice; (2) complement accuracy-focused evaluation with cost-aware and budget-constrained settings; (3) disaggregate results per language rather than reporting aggregate multilingual scores; and (4) analyze downstream effects of tokenization efficiency on cost and environmental impact. For *practitioners*, organizations should audit actual token counts before deployment, evaluate multiple providers (since tokenizer efficiency varies across models), consider language-specific models for high-volume applications, and plan budgets accounting for language-specific variation.

7 Conclusion

This position paper has argued that tokenization is a critical site of linguistic inequity in LLMs. Our empirical analysis of three tokenizers across 20+ languages on three parallel corpora, following the methodology of Rust et al. [2021], Ahia et al. [2023], shows that tokenizer efficiency varies dramatically, with some languages requiring up to $15\times$ as many tokens as English. Drawing on the Sapir-Whorf hypothesis, we introduced *cognitive friction* as an interpretive framework for how misaligned tokenization may degrade not merely efficiency but the quality of model understanding. Synthesizing our findings with the broader literature, we have traced the cascading consequences: a language tax, degraded reasoning, and disproportionate environmental costs (see Appendix I for limitations).

These findings challenge the narrative of AI as a democratizing technology. While it is a remarkable achievement that LLMs *can* be used in many languages, the lack of transparency about lowered performance and higher costs is problematic. The path forward requires transparency about tokenizer efficiency, research into more equitable designs, and policy frameworks that create accountability for linguistic equity. We highlight several open questions for community debate: Should tokenizer efficiency become a standard evaluation metric? How should vocabulary capacity be allocated across languages? Can tokenizers be decoupled from model weights? What role should language communities play in tokenizer design? Is the “English + 1” strategy scalable, or does it fragment the multilingual ecosystem? The question is not whether we can build more equitable language models—we can—but whether we will choose to do so.

References

- Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David R. Mortensen, Noah A. Smith, and Yulia Tsvetkov. Do all languages cost the same? Tokenization in the era of commercial language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 5854–5868, 2023.
- 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. In *Advances in Neural Information Processing Systems*, volume 37, 2024.
- Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Schulze Buschhoff, et al. Tokenizer choice for LLM training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924, 2024.
- Pablo Artola Velasco, Arpit Agarwal, and Krishna P. Gummadi. Is your LLM overcharging you? Tokenization, transparency, and incentives. *arXiv preprint arXiv:2501.00054*, 2025.
- Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. MorphBPE: A morpho-aware tokenizer bridging linguistic complexity for efficient LLM training across morphologies. *arXiv preprint arXiv:2502.00894*, 2025.
- Lera Boroditsky. Does language shape thought? Mandarin and English speakers’ conceptions of time. *Cognitive Psychology*, 43(1):1–22, 2001.
- Iaroslav Chelombitko, Egor Safronov, and Aleksey Komissarov. Qtok: A comprehensive framework for evaluating multilingual tokenizer quality in large language models. *arXiv preprint arXiv:2410.12989*, 2024.
- Alan Cooper, Robert Reimann, and David Cronin. *About Face 3: The Essentials of Interaction Design*. John Wiley & Sons, 2007.
- 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. *arXiv preprint arXiv:2508.04796*, 2025.
- Renato Lui Geh, Honghua Zhang, Kareem Ahmed, Benjie Wang, and Guy Van den Broeck. Where is the signal in tokenization space? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3966–3979, 2024.
- Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, 2021.
- Vani Kanjirang, Tanja Samardžić, Ljiljana Dolamic, and Fabio Rinaldi. 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 24004–24022, 2025.
- Tomasz Limisiewicz, Jiří Balhar, and David Mareček. Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 2350–2367, 2023.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl-2024-00638.
- John A. Lucy. Linguistic relativity. *Annual Review of Anthropology*, 26:291–312, 1997.
- Sachin Pawar, Manoj Apte, Kshitij Jadhav, Girish K. Palshikar, and Nitin Ramrakhiani. Broken words, broken performance: Effect of tokenization on performance of LLMs. *arXiv preprint arXiv:2512.21933*, 2025.

410 Aleksandar Petrov, Emanuele La Malfa, Philip H. S. Torr, and Adel Biber. Language model tokenizers
411 introduce unfairness between languages. In *Advances in Neural Information Processing Systems*,
412 volume 36, pages 38902–38922, 2023.

413 Abrar Rahman, Garry Bowlin, Binit Mohanty, and Sean McGunigal. Towards linguistically-aware
414 and language-independent tokenization for large language models (LLMs). In *Proceedings of the*
415 *IEEE International Conference on Healthcare Informatics*. IEEE, 2024.

416 Nived Rajaraman, Jiantao Jiao, and Ravindran Kannan. Toward a theory of tokenization in LLMs.
417 *arXiv preprint arXiv:2404.08335*, 2025.

418 Souvik Rana, Arul Menezes, Ashish Kulkarni, Chandra Khatri, and Shubham Agarwal. IndicSuper-
419 Tokenizer: An optimized tokenizer for Indic multilingual LLMs. *arXiv preprint arXiv:2511.03237*,
420 2025.

421 Partha Pratim Ray. Does linguistic relativity hypothesis apply on ChatGPT responses? Yes, it does.
422 *Computational Intelligence*, 41(4):e70103, 2025. doi: 10.1111/coin.70103.

423 François Remy, Pieter Delobelle, Hayastan Avetisyan, Alfiya Khabibullina, Miryam de Lhoneux,
424 and Thomas Demeester. Trans-tokenization and cross-lingual vocabulary transfers: Language
425 adaptation of LLMs for low-resource NLP. In *Conference on Language Modeling (COLM)*, 2024.

426 Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. How good is your
427 tokenizer? On the monolingual performance of multilingual language models. In *Proceedings*
428 *of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 3118–3135,
429 2021.

430 Edward Sapir. The status of linguistics as a science. *Language*, 5(4):207–214, 1929.

431 Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with
432 subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational*
433 *Linguistics*, pages 1715–1725, 2016.

434 Jean Seo, Jaeyoon Kim, SungJoo Byun, and Hyopil Shin. How does a language-specific tokenizer
435 affect LLMs? *arXiv preprint arXiv:2502.12560*, 2025.

436 Aaditya K. Singh and DJ Strouse. Tokenization counts: The impact of tokenization on arithmetic in
437 frontier LLMs. *arXiv preprint arXiv:2402.14903*, 2024.

438 Aivin V. Solatorio, Gabriel Stefanini Vicente, Holly Krambeck, and Olivier Dupriez. Double jeopardy
439 and climate impact in the use of large language models: Socio-economic disparities and reduced
440 utility for non-english speakers. *arXiv preprint arXiv:2410.10665*, 2024.

441 Hailay Kidu Teklehaymanot and Wolfgang Nejdl. Tokenization disparities as infrastructure bias: How
442 subword systems create inequities in LLM access and efficiency. *arXiv preprint arXiv:2510.12389*,
443 2025.

444 Oleksii Turuta and Daniil Maksymenko. Tokenization efficiency of current foundational large
445 language models for the Ukrainian language. *Frontiers in Artificial Intelligence*, 8:1538165, 2025.
446 doi: 10.3389/frai.2025.1538165.

447 Chenxi Wang, Yixuan Zhang, Lang Gao, Zixiang Xu, Zirui Song, Yanbo Wang, and Xiuying Chen.
448 Under the shadow of babel: How language shapes reasoning in LLMs, 2025a.

449 Dixuan Wang, Yanda Li, Junyuan Jiang, Zepeng Ding, Ziqin Luo, Guochao Jiang, Jiaqing Liang, and
450 Deqing Yang. Tokenization matters! Degrading large language models through challenging their
451 tokenization, 2025b.

452 Anna Wegmann, Dong Nguyen, and David Jurgens. Tokenization is sensitive to language variation.
453 *arXiv preprint arXiv:2502.15343*, 2025.

454 Benjamin Lee Whorf. *Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf*.
455 MIT Press, 1956.

- 456 Jinbiao Yang. Rethinking tokenization: Crafting better tokenizers for large language models. *Max*
457 *Planck Institute for Psycholinguistics Working Paper*, 2024.
- 458 Mengyu Zheng, Huan Xie, Chaoqun Chen, Jiaqing Ye, Xiang Shen, and Yang Xu. Enhancing large
459 language models through adaptive tokenizers. In *Advances in Neural Information Processing*
460 *Systems*, volume 37, 2024.

461 A Additional Corpus Results

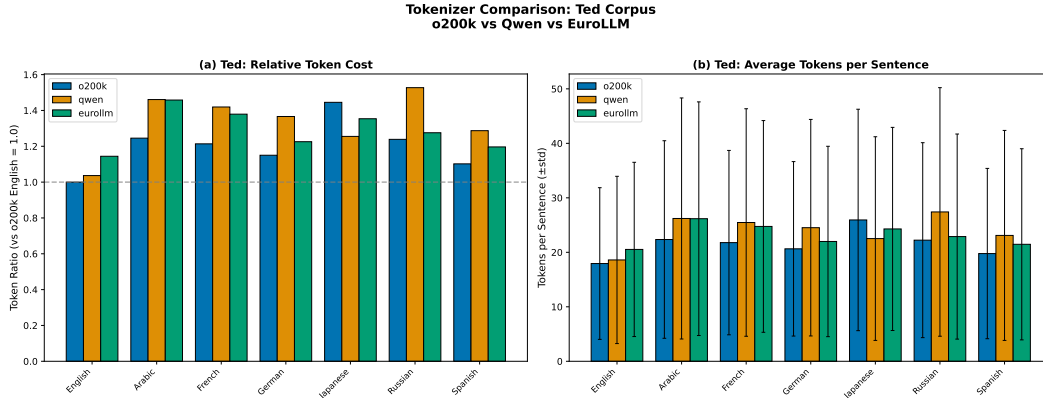


Figure 4: Tokenization efficiency across languages and tokenizers on the TED corpus. **Left:** relative token ratio normalized to English ($1.0\times$); this panel controls for cross-linguistic sentence-length variation. **Right:** absolute tokens per sentence (note: sentence length varies across languages; see left panel for the controlled comparison).

462 B Theoretical Foundations for Tokenization

463 Recent theoretical work by Rajaraman et al. [2025] provides formal justification for why tokenization
 464 is essential for transformer performance. They show that when transformers are trained on data
 465 generated by k -th order Markov processes ($k > 1$), character-level models provably fail under
 466 the assumptions studied, collapsing to predictions equivalent to a unigram character model and
 467 thus incurring high cross-entropy loss. Introducing tokenization over an appropriately constructed
 468 dictionary overcomes this limitation: even simple unigram models over tokens can approximate the
 469 true sequence distribution near-optimally. In particular, Rajaraman et al. [2025, Theorem 4.1]’s main
 470 theoretical result establishes that, with suitable tokenization and a token dictionary of size d , the
 471 achievable cross-entropy loss is bounded within a small multiplicative factor of the optimal loss,
 472 where this factor depends logarithmically on d .

473 C Detailed Empirical Findings on Morphological and Script-Specific Effects

474 **Morphological boundary failures.** Morphological complexity compounds tokenization inequity.
 475 Asgari et al. [2025] demonstrate that BPE’s (Byte Pair Encoding) greedy merging strategy often
 476 fails to align with the actual morpheme boundaries, introducing semantic ambiguity. For example,
 477 the Arabic word *Al-Rahman* (“The Merciful”) may be incorrectly segmented into tokens like *min*
 478 (“whom”), which is semantically unrelated to the original word. Their experiments across four
 479 morphological typologies (English, Russian, Hungarian, Arabic) reveal that standard BPE achieves
 480 near-zero morphological consistency scores for Arabic ($F1 = 0.00$) and English ($F1 = 0.00$),
 481 indicating that frequency-based tokenization systematically destroys morphological structure.

482 **Case-based fertility variation.** Turuta and Maksymenko [2025] provide detailed evidence from
 483 Ukrainian, a morphologically rich language with seven grammatical cases. Their experiments reveal
 484 that tokenization fertility varies systematically across cases: the mean difference from nominative
 485 (base) form ranges from 0.27 to 0.79 additional tokens depending on the model, with dative and loca-
 486 tive cases consistently requiring more tokens than nominative. This demonstrates that morphological
 487 complexity imposes a compounding penalty, with inflected languages requiring more tokens overall,
 488 and each grammatical transformation introduces additional fragmentation.

489 **Domain specialization effects.** Domain specialization compounds tokenization inequity. Turuta
 490 and Maksymenko [2025] demonstrate that tokenization disparities amplify when moving from general
 491 to specialized domains. Their comprehensive evaluation across Ukrainian corpora reveals that while
 492 English shows minimal fertility degradation across domains (~ 0.14 tokens at worst), Ukrainian

493 fertility increases by 0.5–1.0 tokens per word when processing legal texts or scientific abstracts
494 compared to general-domain content. The pattern is stark: for laws, Ukrainian fertility reaches
495 2.00–4.46 tokens per word depending on the model, compared to 1.10–1.14 for English on the same
496 content.

497 **Byte-level tokenizer failures.** Non-Latin scripts suffer disproportionately. Kanjirang et al. [2025]
498 reveal a particularly troubling failure mode: byte-level tokenizers like those used by Llama produce
499 misaligned tokens for non-Latin scripts, interpreting Arabic and Hindi characters as Latin-1 byte
500 sequences. This byte-level fallback not only inflates token counts but fundamentally distorts the
501 input representation, raising questions about whether such models can capture semantic meaning and
502 linguistic nuances at all for affected languages.

503 **Unicode encoding disparities.** Ahia et al. [2024] quantify Unicode encoding disparities: Latin
504 scripts require approximately 1 byte per character, Cyrillic requires 2, and Indic scripts require
505 up to 4 bytes per character under UTF-8. Even with α -sampling during BPE training, which is
506 deemed as a fairness intervention, they find persistent $5\times$ token count disparities between Indic
507 and Latin languages. This demonstrates that data rebalancing alone cannot overcome the structural
508 bias introduced by Unicode encoding asymmetries. Remy et al. [2024] observe that even byte-level
509 tokenizers, which are often presented as a fair alternative, show bias due to the substantial disparities
510 in Unicode encoding efficiency across languages, meaning that no current approach achieves true
511 language neutrality.

512 **Vocabulary coverage gaps.** Many tokenizers lack even complete alphabets for languages spoken
513 by tens of millions of people. Sometimes, the representation is so sparse that basic characters fail
514 to appear in the vocabulary at all Chelombitko et al. [2024]. An analysis of 58 publicly available
515 models revealed that Latin scripts dominate most tokenizer vocabularies (comprising 35–61% of
516 tokens), while scripts serving billions of speakers, e.g., for Cyrillic, Arabic, Greek, receive minimal
517 representation Chelombitko et al. [2024].

518 **Zero-sum allocation.** Tokenization is zero-sum under fixed vocabulary size: a tokenizer with a fixed
519 vocabulary cannot efficiently encode all languages. Vocabulary slots allocated to English morphemes
520 are unavailable for German compounds or Chinese characters. Design choices about which languages
521 to prioritize have direct consequences for all others. Limisiewicz et al. [2023] formalize this through
522 the concept of *vocabulary allocation*, referring to the portion of the multilingual vocabulary devoted
523 to meaningful units of each language. They demonstrate that higher allocation strongly correlates
524 with downstream task performance. The EuroLLM tokenizer seems to confirm this: allocating for a
525 variety of languages leads to mediocre performance for all languages. Ali et al. [2024] quantified
526 this trade-off, finding that multilingual tokenizers covering just five European languages require
527 vocabulary sizes approximately three times larger than monolingual English tokenizers to achieve
528 comparable efficiency.

529 **Parameter budget implications.** The choices made during tokenizer design have profound im-
530 plications for model capabilities. For example, for XLM-RobertaBase approximately 192 million
531 of the model’s 270 million parameters (that is roughly 70%) reside in the input embedding layer
532 [Limisiewicz et al., 2023]. Tokenization determines how efficiently this massive parameter budget is
533 utilized across languages.

534 For an overview of findings across morphological and script-specific studies, see Table 2 in Appendix
535 J.

536 D Detailed Empirical Findings on Cost and Monetary Effects

537 **Mechanism design and incentive misalignment.** Artola Velasco et al. [2025] analyze pay-per-
538 token pricing through a mechanism design lens and demonstrate that this pricing model creates certain
539 incentives: providers could theoretically extract up to three times the fair price by manipulating
540 tokenization, and even under transparency requirements, approximately 13% overcharge remains
541 possible due to information asymmetries. Users cannot verify whether their text was tokenized
542 optimally or whether they are being charged for inefficiencies that benefit the provider.

Tokenizer parity formalization. Petrov et al. [2023] formalize these disparities through the concept of *tokenizer parity*: for any two languages A and B , a fair tokenizer should satisfy $|t(s_A)|/|t(s_B)| \approx 1$, where $t(\cdot)$ denotes the tokenization function and s_A, s_B are parallel sentences. Their analysis reveals that no current tokenizer achieves this ideal. Even for models explicitly targeting non-English languages (German BLOOM, French CroissantLLM, Vietnamese PhoGPT), English remains the closest to parity: the design gravitational pull toward English persists regardless of stated optimization targets. Strikingly, they demonstrate that vocabulary size exhibits diminishing returns: allocating only one-third of the vocabulary to English increases English sequence length by only 10%, suggesting substantial room for reallocation toward underserved languages.

Self-reinforcing cycles and structural inequality. These consequences compound and reinforce each other: English dominance in training data produces English-optimized tokenizers, which make non-English usage more expensive and less effective, which concentrates AI investment and development further in English-speaking contexts. This cycle intersects with existing global inequalities: Ahia et al. [2023] found strong negative correlations between tokenization efficiency and the Human Development Index (HDI) of countries where languages are primarily spoken. Languages spoken in regions with lower HDI (where API costs are least affordable) incur the highest costs. Solatorio et al. [2024] identify a particularly troubling “lower-middle income trap”: speakers in this bracket bear both the largest population affected by tokenization disparities *and* some of the highest premiums, creating a structural barrier to AI adoption precisely where economic development could benefit most.

Concrete cost examples. Ahia et al. [2023] term this “doubled unfairness”: if processing a business email requires 100 tokens in English but 180 tokens in Arabic, the Arabic-speaking user pays 80% more for the same task. At enterprise scale, these differences compound dramatically: serving Arabic, Hindi, or Thai customers costs substantially more than serving English customers, not because those conversations are longer or more complex, but because of tokenizer design choices. In healthcare domains, Rahman et al. [2024] document that Bengali requires approximately 0.82 characters per token versus 4.89 for English under GPT-4’s tokenizer, meaning equivalent clinical content takes nearly six times as many tokens to process.

The 80/20 problem and resource taxonomy. Many “multilingual” models concentrate optimization on a small set of high-resource languages (English, Chinese, Spanish, French) while offering only nominal support for the long tail. Joshi et al. [2021] formalized this disparity through a six-class taxonomy: the 7 “winner” languages (0.28% of the world’s languages) dominate NLP research and resources, while 2,191 languages (88%) are “left behind” with virtually no digital presence. Between these extremes lie “rising stars” and “underdogs”, which are languages with potential but insufficient investment. The result is a two-tiered system: well-served languages for which the model performs as advertised, and poorly-served languages for which users pay premium prices for degraded service. Critically, this taxonomy predicts NLP performance: languages in lower resource classes exhibit systematically worse results on cross-lingual tasks, not because of inherent linguistic difficulty but because of resource poverty. Honest marketing would acknowledge this stratification rather than obscuring it behind aggregate multilingual claims.

For an overview of studies and findings across cost-related studies, see Table 2 in Appendix J.

E Detailed Analysis of Downstream Effects

E.1 Context Window Limitations

Every LLM has a maximum context length, typically measured in tokens. A user working in a token-inefficient language can fit less semantic content into this window, limiting their ability to provide context, include documents, or maintain long conversations. The same 8,000-token context window represents vastly different amounts of usable space depending on language. This has direct consequences for in-context learning: For languages like Telugu and Amharic, examples could not accommodate even a single in-context demonstration, restricting these languages to zero-shot prompting while English users benefit from multiple examples [Ahia et al., 2023]. Teklehaymanot and Nejd [2025] frame this as a fundamental accessibility barrier: languages with Relative Tokenization Cost (RTC) values exceeding 4.0 relative to English effectively receive only one-quarter of the usable

context window, severely limiting their utility for tasks requiring extended reasoning or document analysis.

Chain-of-thought prompting and other reasoning techniques depend on the model maintaining coherent multi-step inference. When the model must process more tokens to represent the same reasoning steps, there are more opportunities for the chain to break. A mathematical proof that fits comfortably in English may strain the model’s coherence when expressed in a more verbose tokenization [Liu et al., 2024].

E.2 Attention Degradation

Transformer models process sequences through attention mechanisms that must track dependencies across all token positions. As sequence length increases, the computational burden grows and the model’s ability to maintain coherence degrades. Research has documented “lost in the middle” effects where information in the center of long contexts is processed less reliably [Liu et al., 2024]. When tokenization artificially inflates sequence length, it pushes more content into these degraded-attention regions, potentially impairing output quality for reasons entirely unrelated to task difficulty. Solatorio et al. [2024] term this phenomenon “double jeopardy”: speakers of low-resource languages face both elevated costs *and* degraded model performance simultaneously. Users pay a premium for inferior service (see Appendices D and G for detailed evidence, and Table 2 in Appendix J).

E.3 Energy Costs and Environmental Impact

Transformer inference scales with sequence length. Processing more tokens requires more compute, more memory bandwidth, and more energy. This means that serving non-English queries is systematically more expensive in terms of infrastructure costs and environmental impact. Solatorio et al. [2024] formalize this relationship: the inference cost in floating point operations scales as $F_{IC} \approx 2PD$, where P is the model’s parameter count and D is the number of tokens processed. Since D is directly inflated by tokenization fragmentation, inefficient tokenization mathematically translates to increased carbon emissions. Empirical analysis reveals that approximately 1.5 billion speakers face tokenization costs 4 to 14 times higher than English speakers, with the carbon footprint of a query depending on the language in which it is asked. The impact during training is even more severe: using English-centric tokenizers can increase computational training costs by up to 68% compared to appropriately designed multilingual tokenizers [Ali et al., 2024].

While individual query differences may seem small, they aggregate to significant environmental cost at scale. If a model serves millions of queries daily across dozens of languages, tokenization inefficiency translates to megawatt-hours of additional energy consumption and corresponding carbon emissions. The environmental burden of AI is disproportionately increased by users who happen to speak tokenizer-inefficient languages.

Climate justice implications. Many of the languages most disadvantaged by English-centric tokenizers are spoken in regions already disproportionately affected by climate change. There is a troubling irony in AI systems that impose higher environmental costs on users from the Global South while those same regions bear the brunt of climate impacts. As noted above, approximately 1.5 billion speakers face tokenization costs 4 to 14 times higher than English speakers, consuming proportionally more energy while delivering inferior results. Tokenization efficiency is thus not merely a technical concern but intersects with broader questions of climate justice and equitable technology access.

E.4 Latency in Time-Critical Applications

These efficiency gaps translate to measurable delays in real-time systems. Healthcare applications demonstrate the severity: medical intake bots and clinical chart summarization are significantly slower for non-English-speaking users, not because the clinical content is more complex, but because the same information requires 2-6 times as many tokens to encode. Rahman et al. [2024] document concrete examples: an LLM-powered medical intake bot may be “significantly slower and less usable for non-English-speaking users” due to inflated token sequences, while physicians in non-English-speaking markets may need to wait longer to summarize a patient’s chart, compared to their English-speaking counterparts. Their analysis confirms the severity: Bengali requires approximately 0.82 characters per token versus 4.89 for English under GPT-4’s tokenizer, meaning equivalent clinical content takes nearly six times as many tokens to process.

647 For an overview of studies and findings across all downstream effects, see Table 2 in Appendix J.

648 **F Detailed Empirical Findings on Cognitive and Linguistic Effects**

649 **Semantic boundary misalignment.** The cognitive consequences of misaligned tokenization extend
650 into the nature of meaning itself. Wang et al. [2025b] demonstrate that Chinese is “more complex
651 and challenging than English in terms of tokenization” (p.3) precisely because it lacks whitespace
652 delimiters. Their adversarial experiments reveal that when Chinese characters form unintended
653 token boundaries (for example, when text meaning “customer raised a glass” is tokenized incorrectly
654 as “client” rather than “customer” + “raised”), LLMs systematically misunderstand the input and
655 produce nonsensical responses, with error rates ranging from 40.91% to 97.73% across models. This
656 demonstrates that tokenization failures cause catastrophic reasoning breakdowns through fundamental
657 misalignment of input representation.

658 **Intra-linguistic variation sensitivity.** Wegmann et al. [2025] demonstrate that tokenization is
659 sensitive to cross-linguistic variation as well as to *intra-linguistic* variation, that is spelling variants,
660 dialectal forms, and stylistic choices within a single language. Their experiments on English dialectal
661 transformations (Appalachian, Chicano, Colloquial Singapore, Indian, and Urban African American
662 English) reveal that models require different tokenizer configurations for tasks that should be *robust* to
663 variation (semantic tasks like classifying the semantic relationship between two text segments) versus
664 tasks that should be *sensitive* to variation (form-based tasks like authorship verification and dialect
665 classification). This finding has profound implications: a tokenizer apparently cannot simultaneously
666 optimize for meaning preservation and stylistic sensitivity.

667 **Linguistic relativity evidence.** Ray [2025] ran a comprehensive quantitative evaluation of linguistic
668 relativity in AI-generated text, testing ChatGPT-4o mini across 13 typologically diverse languages on
669 culturally salient prompts. The results show distinct conceptual and affective output depending on the
670 language used, demonstrating that LLMs translate between languages and impose language-specific
671 conceptual frames on their outputs. Their unsupervised analysis of semantic and sentiment profiles
672 reveals three distinct language groups: Romance languages (Spanish, Portuguese, German) exhibiting
673 high semantic alignment and positive tone; East Asian languages (Chinese, Japanese) showing neutral
674 sentiment and lower cross-linguistic similarity; and a diverse Indo-European cluster (English, Hindi,
675 Arabic, French) with moderate alignment but highest variability. These clusters emerge not from
676 linguistic typology alone but from how the model’s representations interact with each language’s
677 structure, confirming that cognitive friction is not uniformly distributed but follows predictable
678 patterns shaped by training data composition and tokenizer design.

679 **Typologically aligned attention patterns.** Wang et al. [2025a] introduce BICAUSE, a structured
680 bilingual dataset for causal reasoning, and demonstrate that LLMs exhibit *typologically aligned*
681 attention patterns: They find that models internalize language-specific preferences for causal word
682 order and rigidly apply them even to atypical inputs. Chinese reversed causal chains (effect→cause)
683 suffer a 15-percentage-point accuracy drop (76.5% vs. 91.2% for canonical order), while English
684 shows only a 2.5-point drop. This indicates greater flexibility of English causal expression in training
685 corpora. LLMs mimic surface linguistic forms and internalize the reasoning biases shaped by
686 language.

687 **Tokenization parity metrics.** Kanjirang et al. [2025] provide systematic evidence across dialectal
688 NLP tasks, introducing two complementary metrics: Tokenization Parity (TP), which measures how
689 many more tokens a language requires compared to English, and Information Parity (IP), which
690 captures compression efficiency. Their analysis reveals that TP strongly predicts performance on
691 tasks requiring morphological and syntactic features (e.g., extractive QA, where TP-performance
692 correlation reached $r = -0.94$), while IP better predicts semantic tasks like topic classification.
693 These effects are not merely correlational: they reflect the fundamental challenge of processing
694 meaning through a misaligned representational structure.

695 For an overview of studies and findings across cognitive and linguistic studies, see Table 2 in
696 Appendix J.

G Detailed Empirical Findings on Performance and Reasoning Effects

Statistical evidence for tokenization penalty. Tokenization inequity creates measurable performance penalties beyond cost. Pawar et al. [2025] provide rigorous statistical evidence: testing four LLMs across tasks like text classification, math reasoning, and code generation, they demonstrate that when natural words are split into multiple tokens, performance systematically degrades. They call this a “tokenization penalty” for non-English languages.

Causal relationship confirmation. Controlled experiments by Ali et al. [2024] confirm the causal relationship: training identical model architectures with different tokenizers produced performance gaps of up to 9 percentage points on reasoning benchmarks, with the choice of tokenizer alone (not model size, training data, or architecture) accounting for the difference. Zheng et al. [2024] extend this finding by demonstrating that when tokenizer development is coupled with LLM training through iterative refinement, performance improves by over 2 percentage points compared to static frequency-based methods, showing that tokenization choices propagate directly to model capability.

Systematic error patterns. Singh and Strouse [2024] provide evidence that tokenization-induced reasoning failures occur even within English: simply changing whether numbers are tokenized left-to-right versus right-to-left improved GPT-3.5 arithmetic accuracy from 75.6% to 97.8%. Their analysis revealed highly stereotyped error patterns: when input and output token boundaries misalign, the model *always* errs on specific digit positions, demonstrating that tokenization choices create systematic, predictable failures rather than random noise. Crucially, these effects were stronger in smaller models, suggesting that as the field trends toward efficient, compact models, tokenization-induced biases may become more rather than less pronounced.

Universal vulnerability. Wang et al. [2025b] demonstrate that this vulnerability is universal: even the most advanced models fail when tokenization goes wrong. On their adversarial Chinese dataset, error rates ranged from 40.91% (DeepSeek-R1) to 97.73% (Baichuan2-13B-Chat), with GPT-4o at 61.36% and GPT-4 at 50%. Notably, they found that larger models exhibit greater robustness to tokenization errors, but no model is immune.

Model confidence miscalibration. Seo et al. [2025] compared tokenizers for Korean: the English-optimized tokenizers showed higher confidence when making incorrect predictions and produced nonsensical outputs with unwarranted certainty. In comparison, a Korean-extended tokenizer showed lower confidence on errors and lower cross-entropy on complex tasks, indicating more stable and calibrated generation. This demonstrates that cognitive friction manifests not only as degraded accuracy, but as a fundamental miscalibration of model confidence.

For an overview of studies and findings across performance-related studies, see Table 2 in Appendix J.

H Benchmarks Underestimate Real-World Gaps

Standard multilingual benchmarks typically evaluate models on accuracy, fluency, or task completion without accounting for the computational cost of achieving that performance (e.g., MMLU leaderboard¹). A model that scores equally on English and Arabic question-answering may appear equitable, but if it requires four-times as many tokens (and thus quadrupling the cost and compute) to process Arabic, the apparent parity is illusory. Benchmarks should report not only performance metrics but also token counts and efficiency ratios across languages. Recent work has begun to address this gap: Chelombitko et al. [2024] developed Qtok, a framework for systematically evaluating tokenizer quality across languages, analyzing 13 distinct tokenizers across 58 publicly available models. Their analysis revealed that Latin scripts dominate most tokenizer vocabularies (comprising 35–61% of tokens), while scripts serving billions of speakers, e.g., for Cyrillic, Arabic, Greek, receive minimal representation. Such tools represent the kind of infrastructure needed to make tokenization disparities visible and measurable.

Fixed-budget evaluation. A more realistic evaluation paradigm would hold computational budget constant rather than allowing unlimited tokens. Under fixed-budget conditions, the disadvantage of

¹<https://huggingface.co/spaces/StarscreamDeceptions/Multilingual-MMLU-Benchmark-Leaderboard>

inefficient tokenization becomes immediately apparent: models can process less content, maintain less context, and perform fewer reasoning steps for token-inefficient languages. Such evaluations would reveal performance gaps that current benchmarks hide.

User-facing metrics matter. Research benchmarks often diverge from user experience. A user cares about cost per task, latency per query, and quality per dollar, not abstract accuracy on curated test sets. Evaluation frameworks that incorporate these user-facing metrics would better capture the real-world implications of tokenization choices.

I Limitations

This position paper has several limitations that should be acknowledged. First, our empirical analysis focuses on three specific tokenizers (o200k, Qwen, EuroLLM) and may not generalize to all tokenizer architectures. Second, our analysis relies primarily on parallel corpora (FLORES-200, TED, UN), which may not fully capture real-world usage patterns, domain-specific variations, or informal language use. Third, while we document tokenization disparities across 20+ languages, we cannot claim comprehensive coverage of all language families or typological diversity. Fourth, our theoretical framework draws on the Sapir-Whorf hypothesis, which remains debated in linguistics; while we adopt a moderate interpretation, stronger claims about linguistic determinism remain contested. Fifth, tokenization is not the only factor affecting multilingual performance: training data quality, model architecture, fine-tuning strategies, and evaluation methodologies also play crucial roles. Finally, as a position paper, we focus on identifying problems and calling for action rather than proposing specific technical solutions, though we reference promising approaches in the literature. These limitations do not undermine our core argument that tokenization inequity is a critical and understudied problem, but they highlight areas for future research and caution against overgeneralization.

J Papers with Experimental Support for Tokenizer Effects

Table 2: Papers with experimental support for tokenizer effects, focused on diversity of languages, organized by topic focus.

Authors	Main Findings	Context of Analysis	Language Foci
<i>Cost/Monetary Effects</i>			
Ahia et al. [2023]	Documents systematic cost disparities: Telugu and Georgian require up to $5\times$ more tokens than English. Languages like Telugu and Amharic cannot accommodate even a single in-context demonstration due to token inflation.	Cost analysis: API pricing per token creates systematic tax on non-English languages. Context window limitations prevent in-context learning for token-inefficient languages.	Telugu, Georgian, Amharic, and others (multiple languages analyzed)
Teklehaymanot and Nejdil [2025]	Introduces concept of “infrastructure bias”: disparities arise from foundational design choices. Languages with RTC > 4.0 receive only $1/4$ of usable context window. Myanmar script requires $\sim 7\times$ more tokens.	Cost effects: Relative Tokenization Cost (RTC) framework. Context window accessibility barriers. Infrastructure bias analysis.	Myanmar, and others (multiple languages)
Petrov et al. [2023]	Tokenization premium reaches up to $15\times$ for languages like Shan compared to English.	Cost analysis: Commercial API pricing disparities.	Shan (up to $15\times$), and others

Continued on next page

Table 2 – continued from previous page

Authors	Main Findings	Context of Analysis	Language Foci
Artola Velasco et al. [2025]	Pay-per-token pricing creates incentives: providers could extract up to $3\times$ fair price by manipulating tokenization. Even with transparency, $\sim 13\%$ overcharge remains possible.	Cost analysis: Mechanism design analysis of pricing models. Information asymmetry and incentive compatibility.	Not language-specific (pricing mechanism analysis)
Rahman et al. [2024]	Bengali requires $\sim 6\times$ more tokens than English. Healthcare applications show significant latency for non-English users.	Cost effects: Healthcare domain analysis. Latency in time-critical applications.	Bengali ($\sim 6\times$), healthcare applications
Solatorio et al. [2024]	Documents “double jeopardy”: elevated costs AND degraded performance. Languages like Dzongkha and Santali incur costs up to $14\times$ higher than English. Translation accuracy near-zero ($<2\%$) for poorly-served languages vs. $>80\%$ for French/Spanish.	Formalizes inference cost as $F_{IC} \approx 2PD$. Carbon footprint analysis. ~ 1.5 billion speakers face $4\text{--}6\times$ higher costs. Climate justice implications.	Spanish ($1.3\times$), French ($1.3\times$), Chinese ($1.3\times$), Dzongkha (up to $14\times$), Santali (up to $14\times$), and others
<i>Environmental/Energy Effects</i>			
Solatorio et al. [2024]	Inference cost scales as $F_{IC} \approx 2PD$. Carbon footprint depends on language. ~ 1.5 billion speakers face $4\text{--}6\times$ higher environmental costs. Languages like Dzongkha/Santali: $14\times$ higher costs.	Environmental impact: Mathematical formalization of energy costs. Climate justice analysis. Aggregate environmental burden.	Multiple languages (see above)
Ali et al. [2024]	Using English-centric tokenizers increases training costs by up to 68% compared to multilingual tokenizers. Performance gaps up to 9 percentage points on reasoning benchmarks.	Training cost effects: Computational training cost analysis. Performance degradation analysis.	Not language-specific (tokenizer comparison)
<i>Performance/Reasoning Effects</i>			
Pawar et al. [2025]	Rigorous statistical evidence: when natural words split into multiple tokens, performance systematically degrades across text classification, math reasoning, code generation. “Tokenization penalty” for non-English languages.	Performance analysis: Multiple task types. Statistical evidence for degradation.	Multiple languages (non-English focus)
Ali et al. [2024]	Controlled experiments: identical architectures with different tokenizers produce up to 9 percentage point gaps on reasoning benchmarks. Tokenizer choice alone accounts for difference (not model size/data/architecture).	Performance analysis: Controlled experiments. Causal relationship between tokenizer and performance.	Not language-specific (comparative analysis)

Continued on next page

Table 2 – continued from previous page

Authors	Main Findings	Context of Analysis	Language Foci
Zheng et al. [2024]	When tokenizer development coupled with LLM training through iterative refinement, performance improves by >2 percentage points vs. static frequency-based methods.	Performance analysis: Adaptive tokenization methods.	Not language-specific
Singh and Strouse [2024]	Changing number tokenization (left-to-right vs. right-to-left) improved GPT-3.5 arithmetic accuracy from 75.6% to 97.8%. Highly stereotyped error patterns when token boundaries misalign. Effects stronger in smaller models.	Performance analysis: Arithmetic reasoning. Systematic failure patterns.	English (but demonstrates tokenization effects)
Wang et al. [2025b]	Chinese adversarial experiments: error rates 40.91% (DeepSeek-R1) to 97.73% (Baichuan2-13B-Chat). GPT-4o: 61.36%, GPT-4: 50%. Larger models more robust but none immune. Catastrophic reasoning breakdowns from misaligned tokenization.	Performance analysis: Adversarial tokenization challenges. Reasoning failure analysis.	Chinese (Simplified)
Seo et al. [2025]	Korean: English-optimized tokenizers show higher confidence on incorrect predictions. Korean-extended tokenizer: lower confidence on errors, lower cross-entropy on complex tasks. Korean requires $2.6\times$ more tokens.	Performance analysis: Model confidence calibration. Cross-entropy analysis.	Korean ($2.6\times$ multiplier)
Rust et al. [2021]	Translation quality degrades dramatically for non-Latin languages when inefficiently tokenized. Establishes fertility rate metric (tokens per word/character).	Performance analysis: Translation quality. Tokenization efficiency metrics.	Multiple languages (translation tasks)
<i>Morphological/Script-Specific Effects</i>			
Turuta and Maksymenko [2025]	Ukrainian fertility varies by grammatical case: mean difference 0.27–0.79 additional tokens from nominative. Domain specialization compounds inequity: legal texts require 2.00–4.46 tokens/word vs. 1.10–1.14 for English. General→specialized domain: +0.5–1.0 tokens/word.	Performance AND cost effects: Morphological complexity analysis. Domain-specific tokenization penalties.	Ukrainian (morphologically rich, 7 grammatical cases)
Kanjirangat et al. [2025]	Byte-level tokenizers produce misaligned tokens for non-Latin scripts. Arabic and Hindi characters interpreted as Latin-1 byte sequences. Kannada requires $2.19\times$ more tokens. Byte-level fallback distorts input representation.	Performance analysis: Script-specific tokenization failures. Representation bias.	Kannada ($2.19\times$), Arabic, Hindi

Continued on next page

Table 2 – continued from previous page

Authors	Main Findings	Context of Analysis	Language Foci
Ahia et al. [2023]	Unicode encoding disparities: Latin ~ 1 byte/char, Cyrillic ~ 2 , Indic ~ 4 . Even with α -sampling (fairness intervention), persistent $5\times$ token count disparities between Indic and Latin languages.	Performance analysis: Unicode encoding bias. Fairness interventions.	Indic languages ($5\times$), Latin scripts
Remy et al. [2024]	Even byte-level tokenizers show bias due to Unicode encoding efficiency disparities. No current approach achieves true language neutrality.	Performance analysis: Tokenization neutrality analysis.	Multiple languages
Asgari et al. [2025]	BPE’s greedy merging fails to align with morpheme boundaries. Arabic word “Al-Rahman” incorrectly segmented into semantically unrelated tokens. Standard BPE achieves F1=0.00 morphological consistency for Arabic and English.	Performance analysis: Morphological structure preservation. Semantic ambiguity from misalignment.	English, Russian, Hungarian, Arabic
Wegmann et al. [2025]	Tokenization sensitive to intra-linguistic variation: English dialects (Appalachian, Chicano, Singapore, Indian, Urban African American) processed differently. Different tokenizer configurations needed for robust vs. sensitive tasks.	Performance analysis: Dialectal variation. Task-specific tokenization requirements.	English dialects (Appalachian, Chicano, Singapore, Indian, Urban African American)
Yang [2024]	Chinese: $1.7\times$ multiplier. Burmese: up to $10\times$. Amharic: up to $10\times$.	Cost/performance analysis: Tokenization multipliers.	Chinese ($1.7\times$), Burmese (up to $10\times$), Amharic (up to $10\times$)
Chelombitko et al. [2024]	Latin scripts dominate vocabularies (35–61% of tokens). Scripts serving billions (Cyrillic, Arabic, Greek) receive minimal representation. Many tokenizers lack complete alphabets for languages spoken by tens of millions.	Performance analysis: Vocabulary coverage analysis. Script representation disparities.	58 models, 13 tokenizers, multiple languages
<i>Linguistic/Cognitive Effects (Sapir-Whorf Connection)</i>			
Wang et al. [2025b]	Chinese lacks whitespace delimiters, making tokenization “more complex and challenging than English.” Adversarial experiments show that incorrect token boundaries (e.g., “customer raised a glass” tokenized as “client” rather than “customer” + “raised”) cause systematic misunderstanding with error rates of 40.91%–97.73% across models.	Cognitive/linguistic analysis: Adversarial tokenization challenges. Semantic boundary misalignment. Reasoning failure analysis.	Chinese (Simplified)

Continued on next page

Table 2 – continued from previous page

Authors	Main Findings	Context of Analysis	Language Foci
Ray [2025]	Comprehensive evaluation across 13 typologically diverse languages. Distinct conceptual and affective output depending on language. Three language clusters emerge: Romance (high alignment), East Asian (neutral sentiment), Indo-European (moderate alignment, high variability).	Cognitive/linguistic analysis: Linguistic relativity in AI. Semantic clustering.	13 typologically diverse languages
Wang et al. [2025a]	BICAUSE dataset: LLMs exhibit typologically aligned attention patterns. Chinese reversed causal chains: 15-point accuracy drop (76.5% vs. 91.2%). English: only 2.5-point drop. Models internalize language-specific reasoning biases.	Cognitive/performance analysis: Causal reasoning. Language-specific attention patterns.	Chinese, English (bilingual analysis)