
Tokenizer Rankings Reverse Between Arabic and English: ALLaM’s Tokenizer Needs 28% Fewer Arabic Tokens Than Gemma 2’s, 10% More English Tokens

Dmitrii Lozhkin
AVELIN AI · October 26, 2026

Abstract

Retrieved passages are paid for in tokens, and a bilingual corpus pays in two languages. We encoded the same texts with four tokenizers from six open-weight checkpoints (64,000 to 256,000 entries): 915 Arabic and 778 English passages of a corpus of public scientific reports, 2,009 parallel FLORES-200 sentences and 48 complete prompts. The rankings reversed exactly between the languages. ALLaM’s tokenizer needed 28% fewer Arabic tokens than Gemma 2’s (1.67 against 2.30 per word) and 10% more English tokens (1.74 against 1.57), in all 15 Arabic and 9 English documents and on the parallel sentences, where Arabic cost 0.97 times the English tokens under ALLaM and 1.50 times under Gemma 2. Across 17 tokenizer files from 11 lineages, the number of entries containing Arabic letters predicted Arabic tokens per word (Spearman $\rho = -0.99$, also with vocabulary size held fixed); size added no detectable information beyond the entry count. An 8,192-token budget holds 32 Arabic passages under ALLaM and 23 under Gemma 2; in complete prompts the saving was 27% with Arabic evidence and 14% with English evidence.

1 Introduction

A context window is a token budget, and the tokenizer decides how much of it a passage consumes. The same text translated into different languages can differ in tokenized length by up to 15 times [1], and commercial APIs charge speakers of many languages more for poorer results [2]. The standard measure, fertility, counts tokens per word [3].

For Arabic the effect of a dedicated vocabulary is documented. The Jais tokenizer, with 84,992 entries trained on equal Arabic and English text, reports an Arabic fertility of 1.050 against 1.195 for the 250,000-entry BLOOM tokenizer [4]. The released ALLaM 7B checkpoint was trained from scratch with its own Arabic–English tokenizer; vocabulary expansion applies only to the ALLaM models continued from Llama 2 [5]. On an Arabic treebank, the same ALLaM tokenizer needs 1.24 tokens per word against 2.11 for Qwen3 [6]; on our corpus, 1.67 and 2.50.

These studies measure Arabic, usually against one alternative. A bilingual corpus also pays for English, so we measure both languages for four tokenizers on corpus and parallel text, test across 17 tokenizer files whether Arabic entries or vocabulary size predicts Arabic cost, and measure what survives in evidence budgets and complete prompts.

2 Setup

Text. The corpus of public scientific reports holds, without exact duplicates, 915 Arabic passages (138,805 words, 15 documents) and 778 English passages (133,765 words, 9 documents), which are not translations of each other. Parallel text is all 2,009 Arabic–English sentence pairs of the FLORES-200 dev and devtest sets [7, 8] (CC BY-SA 4.0), translated from 562 English Wikinews, Wikivoyage and Wikibooks articles. A measurement set of 30 Arabic and 30 English corpus passages, with 48 Arabic questions and 48 complete prompts, serves the prompt analysis.

Tokenizers. The six checkpoints were Gemma 2 9B and SILMA 9B, which is built on Gemma 2 and ships byte-identical tokenizer files; Qwen3.5 4B and 9B, which share one tokenizer; Falcon-H1 7B; and ALLaM 7B. For the mechanism test we added 13 tokenizer files from Jais family 13B, AceGPT 7B and v2 8B, Fanar 1 9B and 2 27B, Qwen3 8B, Mistral NeMo, DeepSeek-V3, BLOOM, LFM2 1.2B, Phi-4, Granite 3.3 8B and gpt-oss 20B. Vocabularies overlap (all 128,256 Fanar 1 entries occur in Gemma 2’s); merging every pair in which 95% of the smaller vocabulary lies in the larger leaves 11 lineages (15 under a 99% rule). The named models are comparison objects.

Counting and measures. Corpus and parallel texts were normalised to Unicode NFC and encoded without special tokens by each model’s original tokenizer files at a pinned commit; measurement-set and prompt counts come from the tokenizer embedded in each checkpoint’s converted model file. Each complete prompt joins a shared 99-word

Arabic instruction, one or two passages (Arabic in 24 prompts, English in 24) and an Arabic question in the model’s chat template, with special tokens. We report tokens per whitespace-delimited word and ratios of summed tokens. Arabic entries are vocabulary entries whose decoded text contains an Arabic letter. The mechanism test uses Spearman and rank partial correlations across tokenizer files, 100,000-permutation p-values and bootstrap intervals over tokenizers. Other intervals are percentile 95% intervals from 10,000 resamples of documents or, on parallel text, articles.

3 Results

The four tokenizers rank in opposite orders on Arabic and English. On the 915 Arabic passages, the order from fewest to most tokens was ALLaM, Falcon-H1, Qwen3.5, Gemma 2; on the 778 English passages it was exactly the reverse (Table 1, Figure 1a). ALLaM needed 231,580 Arabic tokens against Gemma 2’s 319,732, 28% fewer (ratio 0.724; 95% interval 0.70–0.74), and 232,381 English tokens against 210,609, 10% more (1.103; 1.10–1.11). The full four-way order held in each of the 15 Arabic and 9 English documents, with ALLaM-to-Gemma 2 ratios from 0.63 to 0.77 on Arabic and from 1.07 to 1.14 on English.

Table 1: The four tokenizers. Tokens per word: whole corpus. Parallel: Arabic over English tokens, 2,009 FLORES-200 pairs. Budget: whole Arabic corpus passages in 8,192 tokens, mean of 2,000 random orders.

Tokenizer	Entries	Arabic entries	Arabic tokens per word	English tokens per word	Parallel, Arabic / English	Arabic passages in 8,192 tokens
Gemma 2 / SILMA	256,000	6,119	2.30	1.57	1.50	23.0
Qwen3.5	248,070	8,785	2.03	1.61	1.31	26.2
Falcon-H1	130,049	13,404	1.96	1.69	1.19	27.2
ALLaM	64,000	37,211	1.67	1.74	0.97	32.0

Parallel text repeats the reversal and puts ALLaM at near parity. On the 2,009 translated sentence pairs, ALLaM needed 0.702 times Gemma 2’s Arabic tokens (0.698–0.706) and 1.086 times its English tokens (1.082–1.090), against 0.724 and 1.103 on the corpus, in the same four-way orders. Because each pair says the same thing, the Arabic-to-English token ratio measures what Arabic costs relative to English: 0.97 under ALLaM (0.958–0.974), 1.19 under Falcon-H1, 1.31 under Qwen3.5 and 1.50 under Gemma 2 (1.48–1.51).

Arabic entries predict Arabic cost. A word the vocabulary holds as one entry costs one token; a word it lacks is split into pieces. Among the four, Arabic entries order Arabic cost exactly (ALLaM 37,211 of 64,000 entries, Gemma 2 6,119 of 256,000), while vocabulary size orders them backwards. Across 17 tokenizer files, Arabic entries predicted Arabic tokens per word with $\rho = -0.99$ (95% interval -1.00 to -0.92 ; $p < 0.0001$), and -0.99 with vocabulary size held fixed (partial ρ ; -1.00 to -0.86); on 11 lineages both were -0.99 , and on the 13 added files alone -0.97 (Figure 1b). Size added no detectable information once entries were known (partial $\rho = +0.14$; -0.22 to 0.35); alone, it correlated only weakly ($\rho = -0.38$; -0.85 to 0.23). The smallest vocabulary, AceGPT 7B’s 32,000 entries with 47 Arabic ones, was the least compact (5.50 tokens per word), and the Jais tokenizer (Jais family file), with 43,849 Arabic entries, the most compact (1.58). English does not mirror this: across 17 files, Latin-letter entries rise with vocabulary size ($\rho = 0.88$) and predict English tokens per word no better than size does ($\rho = -0.43$ against -0.42).

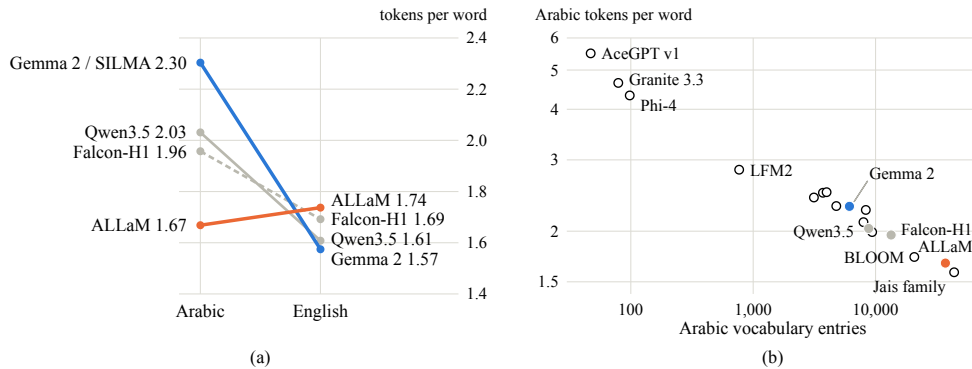


Figure 1: (a) Tokens per word on the whole corpus: the four lines cross between Arabic and English, so each language reverses the other’s order. (b) Arabic tokens per word on the corpus against Arabic vocabulary entries, both on log scales, for 17 tokenizer files; coloured points are the tokenizers of panel (a). Spearman $\rho = -0.99$; against vocabulary size, -0.38 .

Budgets and complete prompts. With 8,192 tokens for passages alone, 32.0 Arabic passages fit under ALLaM (29–35 across random orders) against 23.0 under Gemma 2 (21–26), but 27.0 English passages against 29.9. Complete prompts dilute the difference: ALLaM’s saving was 31% on the measurement-set passages (2,444 against 3,558 Arabic tokens), 27% over the 24 prompts with Arabic evidence (7,417 against 10,114 tokens) and 14% over the 24 with English evidence (8,603 against 9,992), where the Arabic instruction and question keep ALLaM ahead.

The original tokenizer files give the same counts. The converted tokenizers reproduced the original files’ token IDs on all 218 measured strings for Gemma 2, SILMA and ALLaM and on 217 for Qwen3.5; Falcon-H1’s differed on 31, 27 of them splitting a word at an Arabic diacritic. No ranking changed; no passage or prompt total moved by more than 0.51%.

4 Limitations

The corpus is one collection of related public reports (15 Arabic and 9 English documents; one holds 506 of the 778 English passages), and the parallel Arabic is general-domain text translated from English. Per-word figures are not comparable across languages, because Arabic words carry attached clitics; the parallel-text ratio is the cross-language measure. PDF extraction left the article before a hamza- or madda-alef (ﺍ + ﺇ, ﺇ or ﺇ) reversed 81 times against 5,192 correct (1.5%); other lam-alef reversals were not counted. The 17 tokenizer files (11 lineages) are not a random sample and their training data are unobserved, so the Arabic-entry result is an association; the entry count stands in for how much frequent Arabic a vocabulary covers. Answer quality and latency were not measured.

5 Conclusions

Four tokenizers ranked in exactly opposite orders on Arabic and English: ALLaM’s used 28% fewer Arabic tokens than Gemma 2’s and 10% more English ones, in every corpus document and on parallel text, where Arabic cost 0.97 and 1.50 times the English tokens. Across 17 tokenizer files, Arabic entries predicted Arabic cost almost perfectly ($\rho = -0.99$), and vocabulary size added no detectable information once the entries were known. In complete prompts the saving was 27% with Arabic evidence and 14% with English evidence.

6 Recommendations

1. Count each language of a bilingual corpus separately: here a pooled figure would mix a 28% Arabic saving with a 10% English cost.
2. Budget the complete rendered prompt, not the passages: here a 31% passage saving became 27% with Arabic evidence and 14% with English evidence.
3. As a first screen, rank tokenizers by Arabic entries rather than vocabulary size, then measure: the entry count ranked our 17 files with $\rho = -0.99$, an association that does not validate predictions for a new tokenizer.
4. Check converted tokenizers against their original files on the strings to be counted: here one split words at Arabic diacritics on 27 of 218 strings.

Disclosure. The analysis was produced with AI coding agents under the author’s direction. The 17-tokenizer test was fixed before the full run, after a 20-passage smoke run had shown its direction. All numbers were recomputed from per-passage token counts.

References

- [1] A. Petrov, E. La Malfa, P. H. S. Torr, A. Bibi. Language model tokenizers introduce unfairness between languages. In NeurIPS, 2023. arXiv:2305.15425.
- [2] O. Ahia, S. Kumar, H. Gonen, et al. Do all languages cost the same? Tokenization in the era of commercial language models. In EMNLP, 2023. arXiv:2305.13707.
- [3] P. Rust, J. Pfeiffer, I. Vulić, et al. How good is your tokenizer? On the monolingual performance of multilingual language models. In ACL, 2021. arXiv:2012.15613.
- [4] N. Sengupta, S. K. Sahu, B. Jia, et al. Jais and Jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. arXiv:2308.16149, 2023.
- [5] M. S. Bari, Y. Alnumay, N. A. Alzahrani, et al. ALLaM: Large language models for Arabic and English. In ICLR, 2025. arXiv:2407.15390.
- [6] Y. Alakeel, C. Qwaider, H. Aldarmaki, S. Alqahtani. Morphemes without borders: Evaluating root-pattern morphology in Arabic tokenizers and LLMs. In LREC, 2026. arXiv:2603.15773.
- [7] NLLB Team, M. R. Costa-jussà, J. Cross, et al. No language left behind: Scaling human-centered machine translation. arXiv:2207.04672, 2022.
- [8] N. Goyal, C. Gao, V. Chaudhary, et al. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. TACL, 10:522–538, 2022. arXiv:2106.03193.