

BPE vs. Morphology-Aware Tokenization: A Case Study on Arabic Language Modeling at BabyLM Scale

Mohamed Badawy¹ Marion Di Marco^{1,2} Alexander Fraser^{1,2}

¹School of Computation, Information and Technology,
Technische Universität München (TUM)

²Munich Center for Machine Learning

{muhamed.badawy|marion.dimarco|alexander.fraser}@tum.de

Abstract

Byte Pair Encoding (BPE) is blind to morphology, a poor fit for Arabic, whose words bundle conjunctions, articles, and inflection into one token. The leading morphology-aware strategy for Arabic, Farasa+MorphBPE, relies on Farasa, a supervised segmenter disallowed under the BabyLM 2026 rule against off-the-shelf linguistic tools. We study Arabic segmentation under the BabyLM GPT-2 training pipeline and propose **Morfessor+MorphBPE**, an unsupervised morphology-aware tokenizer.

We compare two dimensions: supervised vs. unsupervised *morphological segmentation* and the *merging constraints* used by BPE. All tokenizer variants cluster within a narrow accuracy band, with improvements rather on language understanding tasks than knowledge tasks, even though we do not observe consistent improvements over BPE. Furthermore, we find that holding the Farasa segmentation fixed, the MorphBPE constraint sharply lowers perplexity but barely changes downstream accuracy; and unsupervised Morfessor boundaries match supervised Farasa on aggregate while scoring higher on grammaticality (MultiBLiMP). Morfessor+MorphBPE thus offers a practical path to morphology-aware tokenization for Arabic.

1 Introduction

The BabyLM Challenge asks whether useful language competence can emerge from a developmentally plausible data budget of 100 million words or less (Warstadt et al., 2023; Hu et al., 2024). At this scale, implementation details like tokenization become central; we study Arabic segmentation to optimally represent its complex morphology. Arabic is a natural stress test, since its words bundle conjunctions, articles, and inflection into a single orthographic token (Habash, 2010), so a frequency-driven BPE tokenizer (Sennrich et al., 2016) learns merges that are statistically optimal but morphologically opaque.

MorphBPE (Asgari et al., 2025) addresses this mismatch with a single change to the BPE training loop. Standard BPE repeatedly merges the most frequent adjacent pair of symbols, wherever it occurs; MorphBPE first segments each word into morphemes with an external segmenter and then *forbids any merge whose two symbols lie on opposite sides of a morpheme boundary*. Frequency still decides which merge to apply, but the boundaries act as walls that merges cannot cross, so every learned token stays inside a single morpheme. We refer to this rule as the *MorphBPE constraint*. The MorphBPE recipe for Arabic obtains segmentation boundaries from Farasa (Darwish and Mubarak, 2016), a supervised segmenter trained on the Arabic Treebank. BabyLM 2026 forbids the use of off-the-shelf linguistic tools (Choshen et al., 2026).

In this study, we assess the respective impact of the MorphBPE *constraint* and the *segmentation quality*: we contrast the constrained MorphBPE with plain BPE which allows to merge across morpheme boundaries, and thus is able to compensate over-splitting and to leave high-frequency sequences intact. In addition, we replace Farasa with Morfessor 2.0 Baseline (Creutz and Lagus, 2007; Smit et al., 2014), an unsupervised segmenter trained on simple corpus data.

We compare **Morfessor+MorphBPE** to six reference tokenizers: vanilla BPE, three pure Morfessor variants, Farasa+plain-BPE, and Farasa+MorphBPE. All seven variants cluster within a narrow accuracy band, implying that the unsupervised Morfessor is a sound substitute for Farasa at BabyLM scale. Our contributions are:

1. **Morfessor+MorphBPE** a morphology-aware tokenization strategy for Arabic based on an unsupervised tokenizer competitive with its supervised counterpart.
2. We show a dissociation between the MorphBPE *constraint* and the *segmentation*: with

the Farasa segmentation fixed, the no-cross-boundary constraint sharply improves intrinsic boundary quality and perplexity but leaves downstream accuracy largely unchanged - at this scale, downstream accuracy is robust to both the constraint and the choice of segmenter.

3. While BPE remains strong overall, we find that morphology-aware tokenization yields *task-dependent* effects, with improvements observed rather in language understanding tasks than knowledge or reasoning tasks.

2 Related Work and Background

Tokenization for morphologically rich languages.

A tokenizer maps raw text to the discrete units a language model predicts and is the first step in LLM pre-training. One dominant approach, Byte Pair Encoding (BPE) (Sennrich et al., 2016), starts from characters and iteratively merges the most frequent adjacent pair until a target vocabulary size is reached; merges are driven purely by corpus frequency and are blind to morphology. Previous work has questioned whether standard tokenization approaches like BPE, WordPiece (Schuster and Nakajima, 2012) and SentencePiece (Kudo and Richardson, 2018) can adequately capture morphological structures, and several studies observed that linguistically guided segmentation can lead to improvement in downstream tasks.

Klein and Tsarfaty (2020) argue that linear segmentation is not suitable to capture morphology by investigating the task of multi-tagging Hebrew. Hofmann et al. (2021) find that linguistically aligned segmentation improves sentiment and topic classification.

Further studies evaluate language-specific approaches for morphologically aligned segmentation, for example Antoun et al. (2020) and Asgari et al. (2025) for Arabic relying on Farasa segmentations, and Nzeyimana and Niyongabo Rubungo (2022) relying on a custom morphological tool for Kinyarwanda. Mager et al. (2022) compare morphologically grounded tokenization strategies with BPE for very low-resource languages. In particular, they compare supervised and unsupervised segmentation and find that unsupervised segmentation improves over BPE, whereas supervised morphological segmentation lead to comparatively low results. Hou et al. (2023) explore tokenization with two unsupervised segmentation methods (Morfessor and a finite-state based system) for Russian, Turkish and

Finnish in comparison to BPE, and observe advantages of a morphological segmentation with regard to perplexity, efficiency and performance on downstream tasks. Jabbar (2023) explores non-surface preserving tokenization into a canonical representation requiring an additional detokenization step; he reports improvements on the GLUE Benchmark. Hudspeth et al. (2026) implement a context-aware tokenization strategy for Latin where multiple segmentation options are disambiguated based on POS information.

Fusco et al. (2025) evaluate tokenization strategies in a 10M BabyLM setting. While they find no clear winning tokenizer, with BPE as a strong baseline, they observe that morphological tokenization obtains the best intrinsic segmentation which leads to a more robust generalization in comprehension tasks. This finding is not unsimilar to our general observations.

The BabyLM Challenge and BabyBabelLM.

The BabyLM Challenge (Warstadt et al., 2023; Hu et al., 2024) studies pretraining under human-scale data budgets (10M–100M words), motivated by the observation that human learners acquire language from orders of magnitude less input than production-scale LMs receive. Its 2026 edition expands the challenge to a multilingual setting and introduces an explicit prohibition on off-the-shelf linguistic tools. The training data for self-trained ancillary models is counted towards the total training budget; we follow Fusco et al. (2025) and train our tokenizers on the full corpus.

MorphBPE. MorphBPE is a morphology-aware BPE method (Asgari et al., 2025), which modifies standard byte-pair encoding (Sennrich et al., 2016) by a single algorithmic constraint: pair frequencies are counted only *within* morpheme boundaries provided by an external segmenter, such that no merge can cross a boundary. On Arabic, the paper reports a morphological consistency F1 rising from 0.00 (vanilla BPE at 96K vocabulary) to 0.66 (MorphBPE at 96K). Separately, across four typologically diverse languages, MorphBPE achieves lower language-model training cross-entropy than vanilla BPE.

Two aspects in their evaluation are important for positioning our contribution. First, the reported results are collected at 300M–1B parameter scale on the FineWeb2 corpus, orders of magnitude beyond BabyLM scale, and the authors explicitly leave extrinsic downstream evaluation as future work.

Second, the Arabic experiments use Farasa (Darwish and Mubarak, 2016), a supervised segmenter trained on the Arabic Treebank, disallowed under BabyLM rules. In this paper, we evaluate the difference between Farasa and Morfessor to provide segmentation boundaries for tokenization.

Morfessor and its variants. Morfessor (Creutz and Lagus, 2007) learns morpheme inventories from raw word lists via Minimum Description Length; it is fully unsupervised and requires no external annotation. We use the Baseline model from the Morfessor 2.0 toolkit (Smit et al., 2014), which supports three canonical weighting variants that yield systematically different segmentation behaviour: **Type** (every word type counted once, morphologically faithful), **Token** (log-transformed frequency, a statistical compromise), and **Frequency** (highly frequent words kept unsegmented). All three were compared as tokenizers for English BabyLM in the SIGMORPHON 2024 shared-task submission (Li, 2024), providing the direct precedent for our experiment.

Morfessor was designed for concatenative morphology (Finnish, Turkish), and its behaviour on Arabic’s templatic root-and-pattern morphology (Habash, 2010) is an open question we test empirically. To our knowledge, no prior work has evaluated Morfessor variants as BPE boundary providers for Arabic under BabyLM scale constraints.

3 Methodology

This section describes the data, pre-processing, training pipeline and evaluation.

3.1 Data and preprocessing

All seven tokenizers are trained on the BabyBabelLM Arabic Tier-2 corpus (Jumelet et al., 2025): 30,533 documents, ~ 7.85 M whitespace tokens, and 701,698 unique word types. We use a 95/5 train-validation split, identical across 5 runs with different seeds. We apply *no* Arabic-specific letter normalization (no diacritic stripping, no alef folding); the BabyBabelLM upstream pipeline does not apply such normalization either.

3.2 Seven tokenizers under comparison

The tokenizers span three axes: presence or absence of a morphological prior; whether that prior is unsupervised (Morfessor) or supervised (Farasa); and whether morphological information is used as

the sole segmentation signal (pure Morfessor), as a hard constraint on BPE merges (MorphBPE), or merely as a pre-segmentation that BPE may later merge across (Farasa+plain-BPE).

1. **Vanilla BPE.** A byte-level BPE tokenizer (Sennrich et al., 2016) trained on raw text with frequency-only merges. This is the standard baseline used by the BabyBabelLM Arabic monolingual reference.
2. **Morfessor Type.** Pure Morfessor 2.0 Baseline (Smit et al., 2014) in the Type-weighted variant: every observed word type contributes to the Minimum Description Length objective with equal weight, producing morphologically faithful segmentations.
3. **Morfessor Token.** Pure Morfessor with token-weighted frequency smoothing ($f_{\text{new}} = \log_{30}(f_{\text{old}} + 1)$), a compromise that biases the model toward frequency-typical segmentation.
4. **Morfessor Frequency.** Pure Morfessor with a frequency cutoff: words above a corpus-frequency threshold are left unsegmented, while lower-frequency words are segmented normally. All three Morfessor variants (Type, Token, Frequency) follow the variant taxonomy of Li (2024).
5. **Farasa+plain-BPE.** Farasa’s supervised segmentations are used to pre-segment each word, but plain BPE is then trained on the result with *no* constraint - merges are free to cross the Farasa boundaries. This isolates the effect of the MorphBPE constraint: it shares Farasa’s segmentation with Farasa+MorphBPE but drops the no-cross-boundary rule.
6. **Farasa+MorphBPE.** MorphBPE with Farasa’s supervised segmentations, where merges cannot cross morpheme boundaries (equivalent to the original MorphBPE (Asgari et al., 2025)).
7. **Morfessor+MorphBPE.** Our proposed tokenizer: the MorphBPE algorithm using Morfessor Type as its boundary provider.

3.3 The MorphBPE algorithm

Tokenizers 6 and 7 both instantiate the MorphBPE algorithm, which differs from standard BPE by one line: pair frequencies are counted only *within* morpheme boundaries, so no merge crosses a boundary.

Algorithm 1 (MorphBPE)

1. Initialize vocab with individual chars

2. Segment corpus using morph segmenter
3. while |vocab| < vocab_size:
4. Count byte-pair frequencies
5. [morph-aware: only within-morph]
6. Merge highest-frequency pair
7. Add merged symbol to vocab

The morphological segmenter is invoked only during *training*. At inference, the learned merge table is replayed on the raw character stream as in standard BPE. Because the morpheme-boundary constraint acts only on which merges are allowed to form during training, a merge crossing a morpheme boundary never enters the merge table and can therefore never be applied at inference; the morphological information is baked into the learned vocabulary rather than reapplied per input.

3.4 Boundary providers for MorphBPE

We compare the supervised off-the-shelf tool Farasa and Morfessor, an unsupervised model trained on the BabyLM corpus.

Farasa (supervised). Farasa (Darwish and Mubarak, 2016) is a supervised state-of-the-art Arabic segmenter trained on the Arabic Treebank with hand-annotated morphological segmentation. Crucially, we use Farasa in a *surface-preserving* mode: it only inserts boundaries between characters and never rewrites, expands, or normalizes them, so the concatenation of the segments is identical to the input.

Morfessor Type (unsupervised). Morfessor 2.0 Baseline (Smit et al., 2014) learns a morpheme inventory from raw word-count data via Minimum Description Length, without any external annotation. We train it from scratch on the BabyBabelLM Arabic corpus, restricted to word types with frequency ≥ 30 to filter typos and single-occurrence noise. Type weighting is chosen because it is the most morphologically faithful of the three variants (Section 3.2), which matters when Morfessor’s role is to supply boundaries rather than to segment directly.

Comparison Table 1 shows the different tokenizer outputs on three Arabic words. The contrast is clearest for يكتوبونها (*ya-ktub-ūna-hā*, “they write it”): Morfessor+MorphBPE recovers four morphologically meaningful pieces (prefix, root, plural, object clitic), whereas both Farasa variants keep the verb stem يكتب fused. On يكتبتهم (*bi-maktabati-him*, “in their library”) the three non-Morfessor tokenizers coincide, illustrating that morphology-aware

Tokenizer	Tokenization
المدرسة	<i>wa-l-madrasa</i> , “and the school”
Vanilla BPE	وال مد رسة
Farasa+plain-BPE	وال مد رسة
Farasa+MorphBPE	وال مدرس ة
Morfessor+MorphBPE	و الم درسة
يكتبتهم	<i>bi-maktabati-him</i> , “in their library”
Vanilla BPE	بم كت بت هم
Farasa+plain-BPE	بم كت بت هم
Farasa+MorphBPE	بم كت بت هم
Morfessor+MorphBPE	ب م كت بت هم
يكتوبونها	<i>ya-ktub-ūna-hā</i> , “they write it”
Vanilla BPE	ي كتب ونها
Farasa+plain-BPE	يكتب ونها
Farasa+MorphBPE	يكتب ونها
Morfessor+MorphBPE	ي كتب ون ها

Table 1: Example tokenizations of three Arabic words (| marks token boundaries). Tokens are shown in right-to-left display order; the leading word-boundary marker is omitted. Colour marks morphological quality: **green** = a meaningful morpheme (or a valid fusion of adjacent morphemes, e.g. **وال** = *wa+al*), **red** = a spurious sub-morpheme fragment with no standalone meaning. Morfessor+MorphBPE is fully green on the clitic-heavy verb يكتوبونها but oversplits the stem مكتب in يكتبتهم.

and frequency-driven tokenizers often agree and diverge only on specific constructions.

The examples also reveal a systematic tendency of Morfessor to *oversplit*. On this word, Farasa keeps the noun stem مكتب (*maktab*, “library”) intact - segmenting the word as ب | مكتب | ات | هم - whereas Morfessor+MorphBPE fragments that stem into م | كت | بت | هم, yielding the five-piece ب | م | كت | بت | هم. Because Morfessor is unsupervised and lacks a lexicon of Arabic stems, it sometimes breaks a single morpheme into sub-units that recur frequently in the corpus but are not themselves meaningful. This oversplitting is the main qualitative weakness of the unsupervised boundaries; it is mild on the clitic-heavy verb يكتوبونها, where Morfessor’s prefix/root/suffix split is linguistically apt, but visible on stems that Farasa’s supervised lexicon keeps whole.

3.5 Intrinsic boundary agreement

To quantify the morphological alignment of each tokenizer, we measure segmentation quality with Farasa’s segmentation as reference. For every word in the corpus, we compute precision, recall and F1 of each tokenizer’s splits.

Table 2 reports the result over all word types

Tokenizer	P	R	F1
Vanilla BPE	0.172	0.312	0.222
Farasa+plain-BPE	0.157	0.310	0.209
Farasa+MorphBPE	0.304	0.634	0.411
Morfessor+MorphBPE	0.304	0.682	0.420
Morfessor (pure, Type)	0.334	0.825	0.476

Table 2: Intrinsic boundary agreement with Farasa morpheme boundaries, over all BabyLM-Arabic word types with ≥ 1 reference boundary. (Highest F1 score in bold face.)

that carry at least one reference boundary ($\sim 369K$ types). Because the reference is Farasa, the two Farasa-derived tokenizers could in principle be favoured; in practice they are not near-ceiling (their F1 is around 0.41), because the metric scores the *final* tokenizer output after BPE merges, not the raw segmentation.

Two patterns stand out. First, the *constraint* matters intrinsically: adding the no-cross-boundary rule to the same Farasa segmentation nearly doubles boundary F1 ($0.209 \rightarrow 0.411$), since plain BPE otherwise merges straight back across the morpheme boundaries it was given. Second, the *unsupervised* boundaries are competitive: Morfessor+MorphBPE matches or exceeds Farasa+MorphBPE (0.420 vs. 0.411), and pure Morfessor is the most morpheme-aligned tokenizer of all (0.476), confirming that Morfessor recovers Arabic morpheme boundaries well despite using no labeled data - though it also oversplits, which lowers its precision.

3.6 Implementation choices

We *seed* the vocabulary with a 281-character alphabet that covers ASCII letters, digits, punctuation, Arabic punctuation, and common Unicode symbols, before any morpheme-aware merge is performed. This guarantees that non-Arabic characters have their own tokens, which is necessary for downstream benchmarks whose scoring templates rely on ASCII option labels (A, B, C, D) or digits. Second, we adopt GPT-2’s (Radford et al., 2019) Ġ-prefix word-boundary convention with `add_prefix_space=True`, so that a given word tokenizes identically at every sentence position. Position consistency is a prerequisite for minimal-pair evaluation frameworks that score candidates at arbitrary offsets.

3.7 Model, training, and evaluation

Model. We match the BabyBabelLM Arabic monolingual baseline (Jumelet et al., 2025): a GPT-2 style causal LM (Radford et al., 2019) with 4 transformer layers, 8 attention heads, hidden size 512, FFN size 2048, context length 512, GELU activations, dropout 0.1, and vocabulary size 8,192 - totalling 17.1M parameters.

Training. All seven models are trained for 10 epochs with batch size 16, learning rate 10^{-4} , using five random seeds (42, 41, 5, 11, and 12). Validation loss is tracked on the 5 % held-out split and the checkpoint with best validation loss is used for evaluation.

Evaluation. We use the BabyBabelLM Arabic benchmark suite, scored with a single evaluation harness (lm-eval v0.4.12) and identical task configurations across all tokenizers. The BabyBabelLM Arabic evaluation group (Jumelet et al., 2025) spans minimal-pair grammaticality (MultiBLiMP Arabic), compositional semantics (XCOMPS), natural language inference (XNLI Arabic, MultiNLI Arabic), MMLU-style reasoning, story completion, commonsense reasoning, factual recall, reading comprehension, and topic classification.

4 Experimental Setup

Evaluation harness. All numbers reported in Section 5 are produced by lm-evaluation-harness v0.4.12, invoked with a custom task group `babybabelm_ara` covering the subtasks in Table 3. Scoring is zero-shot with no in-context examples. Each candidate answer is scored by loglikelihood under the model, and the highest-likelihood option is selected. We use `acc, none` as the primary metric for most tasks; TruthfulQA and XStoryCloze use `acc_norm, none` (length-normalised accuracy), following the harness defaults for those tasks. No answer normalization, prompt engineering, or output post-processing is applied.

The BabyBabelLM Arabic task group. Table 3 lists the 14 subtasks retained for analysis; they span formal linguistic competence (MultiBLiMP, XComps), natural language inference (XNLI, MultiNLI), knowledge and reasoning (MMLU, Include-44, BMLama, ARC-Easy, COPA, WinoGrande, TruthfulQA), reading comprehension (Belebele), topic classification (SIB200), and story

Subtask	Type	Metric
MultiBLiMP	Grammaticality (min. pair)	acc
XComps	Compositional semantics	acc
XNLI	NLI	acc
MultiNLI	NLI	acc
MMLU	Multitask knowledge	acc
Include-44	Multi-domain knowledge	acc
BMLama	Factual recall (LAMA)	acc
ARC-Easy	Grade-school science QA	acc
COPA	Causal commonsense	acc
WinoGrande	Coreference / commonsense	acc
TruthfulQA	Truthfulness QA	acc_norm
Belebele	Reading comprehension	acc
SIB200	Topic classification	acc
XStoryCloze	Story completion	acc_norm

Table 3: The 14 subtasks reported for the babybabelm_ara score.

completion (XStoryCloze). The aggregate reported here is the arithmetic mean over the 14 displayed subtask scores. All tasks are sourced from the BabyBabelLM Arabic evaluation suite (Jumelet et al., 2025) and use the task templates distributed with that suite.

5 Results

Table 4 reports accuracy for all seven tokenizers on the 14 BabyBabelLM-Arabic subtasks retained for analysis plus the aggregate score. Per-column best scores are shown in bold, including the best aggregate score.

Aggregate and per-task patterns. On the corpus-level aggregate score, Morfessor Type is the strongest tokenizer (0.3611), followed by Farasa+MorphBPE (0.3377), Morfessor+MorphBPE (0.3361), Vanilla BPE (0.3330), Farasa+plain-BPE (0.3315), Morfessor Token (0.3302), and Morfessor Freq (0.3288). All seven sit within a ~ 3 -percentage-point band, and no tokenizer dominates uniformly across the 14 subtasks. Because the aggregate spread is narrow even with five seeds, we interpret small aggregate differences as *trends* and focus our analysis on the larger, directionally consistent per-task effects and on the controlled pairwise comparisons (constraint on/off; supervised/unsupervised boundaries) that hold all other factors fixed.

Column-wise best scores are distributed across five of the seven tokenizers. Vanilla BPE wins on four subtasks that either scaffold on ASCII characters (ARC-Easy, TruthfulQA) or reduce to lexical overlap (MultiBLiMP, Include-44). Morfessor Type wins on four semantics-/classification-heavy sub-

tasks (XComps, MultiNLI, ARC-Easy tie, SIB200). Morfessor Freq wins on two commonsense reasoning subtasks (COPA, WinoGrande). Farasa+MorphBPE wins on four subtasks that mix NLI, knowledge, reading comprehension, and story completion (XNLI, MMLU, Belebele, XStoryCloze). Morfessor+MorphBPE wins on one factual-recall subtask (BMLama). Farasa+plain-BPE and Morfessor Token win on no subtask.

MorphBPE constraint. Farasa+plain-BPE and Farasa+MorphBPE share an identical Farasa segmentation and differ only in whether BPE merges may cross morpheme boundaries. The constraint (Farasa+MorphBPE) yields a much lower language-model validation perplexity (~ 50 vs. ~ 81), yet the downstream aggregate difference is small (0.3377 vs. 0.3315, ~ 0.6 points). The effect is task-dependent: dropping the constraint helps on MultiBLiMP and COPA but hurts on MultiNLI, Belebele, and XStoryCloze. Thus, the constraint substantially improves the model’s next-token prediction while producing only marginal, mixed downstream effects.

Supervised vs. unsupervised boundaries. At the aggregate level, Farasa+MorphBPE and Morfessor+MorphBPE are essentially tied (0.3377 vs. 0.3361, a ~ 0.16 -point gap). Morfessor+MorphBPE outperforms on MultiBLiMP, XComps, BMLama, and TruthfulQA and underperforms on MultiNLI, COPA, Belebele, and XStoryCloze. The largest single-task gap is on MultiBLiMP, where the unsupervised variant improves grammaticality accuracy by 4.4 percentage points, indicating that the unsupervised boundaries are at least as useful as Farasa’s for the task most directly targeting morphosyntactic competence.

6 Discussion

Validation perplexity. Perplexity is the exponential of the model’s mean per-token cross-entropy on the held-out validation split; lower values indicate a better next-token predictor.¹ We report the three MorphBPE-family tokenizers and pure Morfessor Type in Table 5, where we observe two effects.

¹Perplexity is only loosely comparable across tokenizers, since each induces a different token count for the same text. The three MorphBPE-family tokenizers compared here share vocabulary size 8,192 and a similar segmentation granularity, so their comparison is reasonable but not a normalised per-byte measure. The pure Morfessor Type result is included only as an exploratory reference.

	BabyLM-ara	Grammar & Semantics		Language Understanding				Knowledge & Reasoning						Story	
	Aggregate	MultiBLiMP	XComps	XNLI	MultiNLI	Belebele	STB200	MNLU	Include-44	BMLama	ARC-Easy	COPA	WinoGrande	TruthfulQA	XStoryCloze
Vanilla BPE	0.3330 ± 0.002	0.7539	0.5193	0.3313	0.3166	0.2311	0.1127	0.2307	0.2355	0.1877	0.2586	0.4944	0.4938	0.3333	0.5163
Morfessor Type	0.3611 ± 0.003	0.7531	0.6485	0.3333	0.3590	0.2289	0.2500	0.2295	0.2246	0.1036	0.2586	0.5056	0.4905	0.2058	0.5201
Morfessor Token	0.3302 ± 0.001	0.7416	0.5161	0.3333	0.3244	0.2289	0.1961	0.2295	0.2246	0.1624	0.2556	0.5169	0.4905	0.2857	0.5147
Morfessor Freq	0.3288 ± 0.001	0.7259	0.5226	0.3333	0.3154	0.2289	0.1961	0.2295	0.2246	0.1533	0.2518	0.5281	0.5021	0.2874	0.5170
Farasa+plain-BPE	0.3315 ± 0.002	0.7160	0.5253	0.3325	0.3238	0.2244	0.1961	0.2280	0.2264	0.1740	0.2484	0.5169	0.4773	0.3010	0.4992
Farasa+MorphBPE	0.3377 ± 0.003	0.7004	0.5273	0.3353	0.3396	0.2456	0.1961	0.2342	0.2192	0.1857	0.2548	0.4831	0.4946	0.2772	0.5217
Morfessor+MorphBPE	0.3361 ± 0.001	0.7449	0.5333	0.3333	0.3168	0.2333	0.1961	0.2307	0.2264	0.1888	0.2488	0.4607	0.4938	0.2925	0.4992

Table 4: BabyBabelLM Arabic evaluation across the seven tokenizers, with the 14 subtasks grouped into four capability categories (Grammar & Semantics; Language Understanding; Knowledge & Reasoning; Discourse). **BabyLM-ara** is the official babybabel1m_ara aggregate; it is the benchmark-defined score and is *not* the arithmetic mean of the displayed subtask columns. Per-subtask scores are single-seed (42). Per-column best is bolded.

Tokenizer	Val. PPL ↓
Morfessor Type	50.7
Farasa+plain-BPE	80.9
Farasa+MorphBPE	50.2
Morfessor+MorphBPE	40.5

Table 5: Validation perplexity (lower is better). Adding the MorphBPE constraint to a fixed Farasa segmentation lowers perplexity by ~38 % (80.9→50.2); replacing Farasa’s supervised boundaries with unsupervised Morfessor boundaries lowers it by a further ~19 % (50.2→40.5). Pure Morfessor Type is included as an exploratory reference and is not used in the controlled comparisons.

The *constraint* effect: adding the no-cross-boundary rule to a fixed Farasa segmentation cuts perplexity from 80.9 to 50.2. The *supervision* effect: replacing Farasa’s supervised boundaries with Morfessor’s unsupervised ones cuts it further to 40.5 - so the rule-compliant, unsupervised tokenizer is in fact the strongest next-token predictor of the three matched MorphBPE-family configurations. The exploratory pure Morfessor Type result (50.7) is higher than that of Morfessor+MorphBPE (40.5), suggesting that Morfessor segmentation alone does not explain the latter’s low perplexity.

Perplexity and downstream accuracy are not connected. The clearest single result is that the MorphBPE constraint buys a large reduction in validation perplexity (~50 vs. ~81 against its unconstrained twin) but only a ~0.6-point aggregate change downstream. Because the two tokenizers being compared share an identical Farasa segmentation, this is not a confound of segmentation quality: the constraint genuinely helps the language model predict the next token, yet that intrinsic gain does

not propagate to the multiple-choice and minimal-pair tasks that make up the benchmark suite. The practical lesson for BabyLM-scale tokenizer design is that validation perplexity is a poor proxy for downstream utility, and tokenizers should be selected on task metrics rather than on training loss.

At this scale, segmentation matters more than the constraint. Reading the three Farasa-segmented tokenizers together - vanilla BPE (no Farasa), Farasa+plain-BPE (Farasa segmentation, no constraint), and Farasa+MorphBPE (Farasa segmentation, with constraint) - the downstream scores move little once the segmentation is fixed. The no-cross-boundary rule reshapes the learned merge table and the model’s perplexity, but the units the model ultimately sees are dominated by the segmentation it was given. This suggests that, at 10M tokens, the value of morphology-aware BPE is carried mostly by *where the boundaries are drawn*, not by *whether merges are forbidden from crossing them*.

Unsupervised boundaries are competitive with supervised ones. Holding the MorphBPE constraint fixed, replacing Farasa’s supervised boundaries with Morfessor’s unsupervised ones costs essentially nothing on aggregate (0.3377 vs. 0.3361) and *improves* the benchmark most directly tied to morphosyntax (MultiBLiMP, +4.4 points). Our experiments indicate that not relying on a supervised segmenter does not degrade morphological competence at this scale. Morfessor+MorphBPE is thus not a compromise, but is competitive on its own terms.

Gains live on formal, not functional, competence. Every tokenizer performs near chance on the knowledge- and reasoning-oriented subtasks (MMLU, BMLama, ARC-Easy, Belebele), and tokenizer choice barely moves them. Where tokenizers separate at all is on minimal-pair grammaticality and compositional semantics. In the formal-vs-functional framing of Mahowald et al. (2024), our results place any benefit of morphology-aware tokenization squarely on the *formal linguistic competence* axis: it can help a small model learn the shape of the language, but it cannot manufacture world knowledge that the 10M-token budget never contained. This is consistent with prior BabyLM findings that knowledge tasks are largely unlearnable at this scale (Jumelet et al., 2025; Hu et al., 2024), and it clarifies what a tokenizer intervention can and cannot be expected to do.

7 Conclusion

We studied morphology-aware tokenization for Arabic at BabyLM scale through a controlled comparison of seven tokenizers - vanilla BPE, three pure Morfessor variants, and three Farasa/Morfessor combinations - all trained and evaluated under an identical pipeline. Overall, BPE remains a strong baseline and no tokenizer emerges as a clear winner.

Our central contribution is **Morfessor+MorphBPE**, an unsupervised morphology-aware BPE tokenizer that, to our knowledge, is the first Arabic morphology-aware tokenizer compliant with the BabyLM 2026 prohibition on off-the-shelf linguistic tools. It is competitive with its supervised counterpart on aggregate accuracy and matches or exceeds it on minimal-pair grammaticality, without consulting any labeled resource. Two comparisons provide further insights: holding the Farasa segmentation fixed, the MorphBPE no-cross-boundary constraint substantially lowers perplexity but barely changes downstream accuracy, indicating that at this scale the segmentation - not the constraint - drives downstream behaviour; and holding the constraint fixed, unsupervised Morfessor boundaries are competitive with supervised Farasa boundaries. Any benefit of morphology-aware tokenization concentrates on formal linguistic competence rather than on knowledge-based tasks, which remain near chance at 10M tokens. We hope these results offer a practical, rule-compliant default for tokenizing low-resource morphologically rich languages, and a note of caution against selecting tokenizers by

perplexity alone.

Limitations

Limited seed budget. Every model is trained with five random seeds (42, 41, 5, 11, and 12). This improves stability over a single run, but the aggregate scores still fall within a ~ 3 -percentage-point band, so we deliberately frame small aggregate differences as trends (Section 5). The larger per-task effects (e.g. the ~ 4 -point MultiBLiMP gap and the perplexity gap between the constrained and unconstrained Farasa tokenizers) are more likely to be robust, but broader replication would be needed to establish fine-grained significance.

Single corpus, scale, and architecture. All experiments use the BabyBabelLM Arabic Tier-2 corpus (~ 10 M tokens), one vocabulary size (8,192), and one GPT-2-style architecture (17.1M parameters). We do not test whether the observed patterns hold at the 100M-token tier, at larger vocabularies (the original MorphBPE paper uses 96K), or under a masked or GPT-BERT objective. The near-chance performance on knowledge and reasoning tasks is a property of this scale, not of any tokenizer.

Morfessor and templatic morphology. Morfessor was designed for concatenative morphology and segments Arabic by splitting the linear character string; it cannot represent the non-concatenative root-and-pattern structure of Arabic templatic forms. It nonetheless reliably captures Arabic’s frequent linear affixation - prefixed conjunctions and prepositions, and suffixed inflection and object clitics - directly from raw word counts and without any labeled segmentation data, which is exactly the boundary information our morphology-aware tokenizer relies on. We explore only its three standard Baseline variants and do not evaluate Morfessor’s EM/Categories models or other unsupervised segmenters.

No intrinsic morphological metric. We report downstream task accuracy but not the intrinsic morphological consistency metrics (μ_c , μ_e) used by the original MorphBPE paper, so we cannot directly relate our downstream results to tokenizer-level morphological fidelity. We also did not audit Farasa’s segmentation errors, which propagate into the two Farasa-based tokenizers.

Evaluation scope. We evaluate the models only on the official BabyBabelLM Arabic benchmark

suite used in the BabyLM evaluation. We do not test them on additional external Arabic downstream tasks or generation benchmarks. Our conclusions are therefore limited to the tasks represented in the official suite.

Acknowledgments

This work was co-funded by the European Union (ERC, EPICAL, 101141712). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. [AraBERT: Transformer-based model for Arabic language understanding](#). In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools*.
- Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. 2025. [MorphBPE: A morpho-aware tokenizer bridging linguistic complexity for efficient LLM training across morphologies](#). *arXiv preprint*.
- Leshem Choshen, Alex Warstadt, Aaron Mueller, Candace Ross, Ryan Cotterell, Tal Linzen, and Ethan Gotlieb Wilcox. 2026. [BabyLM turns 4 and goes multilingual: Call for papers for the 2026 BabyLM workshop](#). *Preprint*, arXiv:2602.20092.
- Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1).
- Kareem Darwish and Hamdy Mubarak. 2016. [Farasa: A new fast and accurate Arabic word segmenter](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*.
- Achille Fusco, Maria Letizia Piccini Bianchessi, Tommaso Sgrizzi, Asya Zanollo, and Cristiano Chesi. 2025. [Linguistic units as tokens: Intrinsic and extrinsic evaluation with BabyLM](#). In *Proceedings of the First BabyLM Workshop*, pages 496–507, Suzhou, China. Association for Computational Linguistics.
- Nizar Y. Habash. 2010. *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.
- Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. [Superbizarre is not superb: Derivational morphology improves BERT’s interpretation of complex words](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3594–3608, Online. Association for Computational Linguistics.
- Jue Hou, Anisia Katinskaia, Anh-Duc Vu, and Roman Yangarber. 2023. [Effects of sub-word segmentation on performance of transformer language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7413–7425, Singapore. Association for Computational Linguistics.
- Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. [Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora](#). In *Proceedings of the 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*.
- Marisa Hudspeth, Patrick J. Burns, and Brendan O’Connor. 2026. [Contextual morphologically-guided tokenization for Latin encoder models](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5761–5775, Rabat, Morocco. Association for Computational Linguistics.
- Haris Jabbar. 2023. [MorphPiece: A linguistic tokenizer for large language models](#). *arXiv preprint arXiv:2307.07262*.
- Jaap Jumelet, Bastian Bunzeck, Manar Ali, Mohammad Amin Ghanizadeh, Laurent Prevot, Sheng-Fu Wang, Jou-An Chi, Shu-Kai Hsieh, Francesca Padovani, Alex Warstadt, Sina Zarrieß, and Ryan Cotterell. 2025. [BabyBabelLM: A multilingual benchmark of developmentally plausible training data](#). *arXiv preprint*.
- Stav Klein and Reut Tsarfaty. 2020. [Getting the ##life out of living: How adequate are word-pieces for modelling complex morphology?](#) In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 204–209, Online. Association for Computational Linguistics.
- Taku Kudo and John Richardson. 2018. [SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.
- Zilong Li. 2024. [Team Ryu’s submission to SIGMORPHON 2024 shared task on subword tokenization](#). *arXiv preprint*.
- Manuel Mager, Arturo Oncevay, Elisabeth Mager, Katharina Kann, and Ngoc Thang Vu. 2022. [BPE vs. morphological segmentation: A case study on machine](#)

translation of four polysynthetic languages. In *Findings of the Association for Computational Linguistics: ACL 2022*.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 28(6):517–540.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Mike Schuster and Kaisuke Nakajima. 2012. [Japanese and korean voice search](#). *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

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

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. [Morfessor 2.0: Toolkit for statistical morphological segmentation](#). In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. [Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*.

A Hyperparameters

Table 6 lists the hyperparameters used for all experiments. The model architecture and training recipe are identical across all seven tokenizers; only the tokenizer varies.

Model (GPT-2)	Value
Transformer layers (n_{layer})	4
Attention heads (n_{head})	8
Hidden size (n_{embd})	512
FFN size (n_{inner})	2048
Context length	512
Activation	gelu_new
Dropout (attn / embd / resid)	0.1
Initializer range	0.02
LayerNorm ϵ	10^{-5}
Vocabulary size	8,192
Total parameters	17.1M
Training	Value
Epochs	10
Batch size	16
Learning rate	10^{-4}
Max sequence length	512
Validation split	5 %
Dataset	BabyLM-community/ babyLM-ara
Tokenizer	Value
Vocabulary size (all)	8,192
Min frequency (BPE)	2
Arabic normalization	None
Add-prefix-space	True (GPT-2 ğ)

Table 6: Hyperparameters shared across all tokenizer variants.