

Tokens of Meaning: Semantically-Informed Morphological Tokenization in Low-Resource Language Modeling

Juan Diego Rodriguez
Department of Computer Science
The University of Texas at Austin
juand-r@utexas.edu

Abstract

Subword tokenizers for language models are typically trained to efficiently encode text into short sequences of reusable units. These often produce tokens which are not linguistically meaningful, such as `giraffe`→`g` `ir` `affe`. We ask whether a tokenizer that instead splits words into semantically meaningful morphemes can help a language model learn under a developmentally plausible data budget, where every training token is scarce and sharing evidence across related word forms should matter most. We build such a tokenizer for English by hand-curating a word-to-morpheme-segmentation dictionary, built on top of MorphyNet, a database of derivational and inflectional morphology. Each word is only split into morphemes when the pieces are productive and the whole word’s meaning is semantically related to them, thus splitting `unhappiness`→`_un` `happy` `ness`, but keeping `business` whole. On the 10M-word BabyLM corpus this holds out-of-vocabulary tokens to 1.0%, with an inventory of ~37k tokens. We compare the tokenizer intrinsically against both statistical (BPE, Unigram, WordPiece) and morphology-aware (Morfessor, MorphPiece) baselines, and measure its effect on a child-scale language model. In a controlled comparison against an uncured baseline built from the same morphological source, curation yields +1.4 BLiMP. Our submitted model, which adds masked next-token prediction and vocabulary pruning, reaches 72.53 BLiMP and ranks second on COMPS and third on EWok on the BabyLM 2026 strict-small leaderboard.¹

1 Introduction

Remarkably successful though LLMs may be at producing fluent and felicitous language, they require orders of magnitude more training data than

a human encounters in a lifetime. A child, on the other hand, acquires the ability to communicate in their native language from an exposure to only tens of millions of words (Warstadt et al., 2023). This gap in data efficiency is the motivation behind the BabyLM Challenge (Warstadt et al., 2023), a competition on language modeling with a developmentally-plausible budget (10M or 100M words). Recent work at this scale finds that tokenization is not a neutral preprocessing step, but can also yield gains on par with other modeling choices (Alqahtani et al., 2026; Altintas et al., 2025). Reduced or restructured vocabularies change what a small model can learn, and can yield improvements in downstream tasks (Muckatira et al., 2024; Finke et al., 2025).

We take inspiration from the importance of morphology in human language learning and use. Children readily recognize and extend morphological regularities. Shown a nonce word, they inflect it without hesitation (one wug, two wugs; Berko, 1958). Knowing *shelter*, a learner also has *sheltering*, *sheltered*, and *shelters*; and we grasp that *slithy toves* denotes more than one *tove* that is *slithy*, though we have seen neither word before. A language model could exploit the same structure. With a 10M-word budget, where exposure to any single word form is thin, **related words can draw strength from the morphemes they share**. Every occurrence of *walked* or *walking* also reinforces *walk*. We build this inductive bias directly into the tokenizer.²

Standard subword tokenizers do not do this. BPE (Sennrich et al., 2016), WordPiece (Wu et al., 2016), and Unigram (Kudo, 2018) are all trained to compress a corpus under a vocabulary size bud-

¹Our code is available at <https://github.com/juand-r/morpheus-tokenizer>.

²We do not model the learning of morphological segmentation; we treat morphology as given and ask how much is gained when the learner starts with it. Bölücü and Can (2025) instead experiment with a curriculum that gradually introduces morphological segmentation during training, with mixed, task-dependent results.

get, and the segments they induce often disregard morpheme boundaries. The result is segments with no semantic meaning and that fail to align related words: a purely statistical split may render *adder* as *ad der*, split *singleton* as *sing leton*, or tokenize *unhappiness* and *happiness* so that they have no tokens in common.

A growing body of work documents the downstream cost of this morphological blindness, for derivational generalization (Hofmann et al., 2021, 2022), for the reliability of model behavior (Rust et al., 2021), and for morphologically-rich languages (Schwartz et al., 2020). When a tokenizer is made morphology-aware the payoff is largest in the data-scarce and morphologically-rich regimes—from child-scale BabyLM models (Bölücü and Can, 2025; Fusco et al., 2025) to Kinyarwanda (Nzeyimana and Niyongabo Rubungo, 2022), Hebrew (Gueta et al., 2023), and the morphologically complex languages of Asgari et al. (2025).

How much does a small language model gain when its tokenizer performs morphological splits that are both **productive** and **semantically related**? It is not enough to split on morpheme boundaries: an aligned-but-meaningless split (*adder*) injects a false training signal, teaching the model that *adder* (Vipera berus, a snake) is a kind of adding, and polluting the embeddings of both pieces. A useful morphological split must satisfy two conditions: (1) the pieces must be genuinely reused across the lexicon (*productive*), and (2) the pieces must be semantically related to the word they build. We maximize how much evidence morphemes pool, but only over segmentations that could potentially help the model learn new meanings. A prior BabyLM submission pursued morphological tokenization but did not enforce this constraint (Bölücü and Can, 2025).

Beyond the modeling gain, there is an independent, computational-linguistic reason to want morphologically correct tokenization: it produces tokens that are morphemes, making the model’s vocabulary interpretable and its behavior on morphological phenomena easier to analyze (Hofmann et al., 2021; Batsuren et al., 2022). A tokenizer whose units are real morphemes provides a useful instrument for studying morphological phenomena with language models.

Contributions As our entry to the BabyLM 2026 Challenge, this paper contributes: (1) an

explicit, documented criterion for when and how to split a word into morphemes—productivity *and* semantic compositionality—stated formally enough to expose which parts could be automated (Section 3); (2) a publicly-available, hand-curated, lemma-based morphological tokenizer for English built on this criterion over MorphyNet (Batsuren et al., 2022), which holds corpus OOV to 1.0% with no subword fallback while collapsing related forms onto shared stems; (3) an intrinsic comparison of this tokenizer against five automatic and semi-automatic tokenizers (BPE, Unigram, WordPiece, Morfessor, MorphPiece) on identical data (Section 5.1); (4) a controlled downstream study (fixing the model and data, only varying the tokenizer) of the tokenizer’s effect on a child-scale language model (Section 5.2); and (5) our BabyLM 2026 competition submission (Section 5.3).

Our submitted model (Morpheus)³ ranked second on COMPS (54.67), third on EWok (53.2), seventh on BLiMP (72.53), and seventh on GlobalPIQA (38.65) on the BabyLM 2026 strict-small leaderboard. We release the tokenizer and every comparison artifact.

2 Related Work

Standard subword tokenizers such as BPE (Sennrich et al., 2016), WordPiece (Wu et al., 2016) and Unigram (Kudo, 2018) optimize compression, not morphology. Unigram aligns with morphology better than BPE (Bostrom and Durrett, 2020; Vemula et al., 2025), yet it still fragments words across meaning (*singleton*→*single t on*) and never unifies an inflectional paradigm onto one unit. Morphology-aware tokenizers instead segment on morphemes. However, unsupervised (Morfessor; Smit et al., 2014), lexicon-plus-fallback (MorphPiece; Jabbar, 2023), decode-time (FLOTA; Hofmann et al., 2022), merge-constrained (MorphBPE; Asgari et al., 2025), or byte-level (MYTE; Limisiewicz et al., 2024) all apply morphology automatically. Closest to us, two BabyLM submissions (Bölücü and Can, 2025; Fusco et al., 2025) train child-scale models on automatically segmented vocabularies with no semantic constraint. We argue that *how* the split is chosen, not merely that morphology is used, should matter. Gains from morphology-aware to-

³<https://huggingface.co/juand-r/morpheus-10M-v5>

kenization recur across morphologically rich languages (Toraman et al., 2023; Nzeyimana and Niyongabo Rubungo, 2022; Gueta et al., 2023; Schwartz et al., 2020). We give the full discussion of related work in Appendix E.

3 Problem Formulation

Given a word, do we decompose it into morphemes or keep it atomic, and if we split, how? Here we state our procedure, both to document the method and to expose which parts could be automated. We applied this procedure semi-manually, assisted by Claude Code and always verifying the results.

3.1 The objective

We want our tokenizer to serve a model that learns one embedding per morpheme, such that a word’s meaning and distribution should be **related to, and shared with, its pieces**. A split is good exactly when that sharing helps the model and bad when it misleads it. Two forces are in tension:

- **Economy / reuse** (push to split): decomposing onto shared morphemes shrinks the vocabulary and lets evidence for a morpheme pool across every word containing it (e.g., *seem* gets evidence from *seemed/seeming/seems*).
- **Semantic integrity** (push to keep whole): a split forces the model to explain the word from its pieces. If the pieces do not carry the word’s meaning, the split injects a noisy training signal; e.g., it teaches the model that *business* is a kind of *busy-ness*, polluting *busy* with associations with commerce it should not carry.

For a word w with a candidate segmentation $s(w) = [m_1, \dots, m_n]$ ($n \geq 2$), we accept the split if (P) every piece is productive and (S) the pieces are semantically related to the word; otherwise we keep w atomic. When more than one valid segmentation exists, we take the one that pushes as much of the word onto shared morphemes as the two conditions allow. We maximize morpheme reuse, but only over segmentations whose every piece is a real shared unit and whose pieces stay related to the word’s meaning. Metrics that reward raw boundary alignment (MorphScore Arnett and Bergen, 2025) or subword productivity, such as the reuse measure ρ used by Parra (2024), optimize the sharing objective without the constraint.

Condition (P): productivity / reuse Every piece m must be a productive morpheme: genuinely shared, not a fragment invented for one word. This can happen in the following three ways:

- (1) m is a closed-class **affix** in the recognized inventory (un-, -ness, -ic, -ion, ’s, ...), or
- (2) m is a **free word** (attested standalone), or
- (3) m is a **bound root that recurs** — it appears as a piece in at least 2 other words’ segmentations: *nutrit* recurs across *nutrition/nutritious/nutritive*; *mester* across *semester/trimester*. A piece used for a single word fails.

A piece used in exactly one word buys no sharing; it just fragments a word into *fragment* + *affix* and adds a dead vocabulary row.

Condition (S): semantic relatedness The pieces must be genuinely related to the word’s meaning, so that the split does not create a false or misleading signal. There are four ways (S) can fail:

S1 — Unrelated meaning. Is w ’s dominant modern sense genuinely related to the pieces’ senses? *crier* = one who cries (✓); *suburbia* = the realm of suburbs (✓); but *business* is not the state of being busy, so we keep it atomic, and likewise for *legitimate*, *livery*, *apartment*.

S2 — False etymology. The pieces may be real morphemes yet the connection is coincidental, not synchronically true: *adder* is a snake, not add er; *polish* is not pole ish; *glower* is not glow er.

S3 — Incorrect homonym sense. If the base is a homonym (one form, unrelated senses), split only when the sense the word uses is the one the shared token would usually carry: *lime-stone* is kept atomic because its *lime* is the mineral, not the far more common citrus fruit.

S4 — Pollution by collision. If a piece is spelled like a much more frequent, unrelated token, splitting pours w ’s semantics into it: *werewolf* is not were wolf (the morpheme *were-* collides with the copula *were*); *singleton* is not single ton (diminutive *-ton* collides with the weight unit). We keep these atomic.

S1–S4 are semantic judgments. These judgments are our contribution over automatic morphological

Trigger	Fails	Examples (kept atomic)
Lexicalized / unrelated meaning	S1	<i>business, legitimate, livery</i>
False etymology (real pieces, wrong sense)	S2	<i>adder, polish, tally, glower, carter</i>
Dead / deep etymology (only historical)	S2	<i>werewolf, remember</i>
Wrong-sense homonym	S3	<i>limestone</i>
Collision with frequent unrelated token	S4	<i>singleton (ton), werewolf (were)</i>
Phantom / non-recurring piece	P	<i>cranberry</i>
Proper name, common-noun use less frequent	entry	<i>hayward, hooper</i>
Genuine ambiguity	entry	<i>leaves</i>

Table 1: Keep-atomic triggers, indexed to the failing condition, with real cases decided during curation.

tokenizers. Possible automatic signals such as distributional relatedness in embedding space (*business* sits far from *busy*), definitional/gloss overlap, and frequency-collision checks may help reduce some of the annotation work in the future.

3.2 Decision procedure

Figure 1 shows the decision procedure. A candidate split must satisfy every condition in order to be accepted.

Two special cases keep a word atomic before these checks: **genuine ambiguity**, where a form has two valid analyses that discard information (*leaves* = *leaf*+s or *leave*+s); and **proper names** with no common-noun use in the corpus (*hayward*, *hooper*), whose split would attach name occurrences to unrelated common words.

3.3 How to split, once accepted

Once a split is accepted, we use a few conventions to normalize it. We emit canonical, lemma-level morphemes rather than surface substrings, treating orthographic realization as a separate layer in order to collapse spelling variants (e.g., *-ise/-ize*, British/American forms, elisions) onto one shared morpheme. The same convention absorbs *allomorphy* (the different surface shapes a single morpheme can take) so for example the plural allomorphs *-s/-es/-ies* map to one *-s*, and the negative prefix *in-/im-/il-/ir-* to one *in-*. A model that would otherwise spread its evidence across each variant instead treats these as a single unit. Beyond spelling, we prefer a free word or a recurring root to an invented bound stem (keeping *legitimate* whole rather than minting *legitim*,

but allowing *nutrit* and *epidem* because they recur across a family); and we drop a meaningless linking element (interfix) rather than tokenize it (e.g., *cosmonaut* → *cosm naut*, dropping the *-o-*), except where dropping it would cause a collision.

4 Method

We fix one model architecture, one training corpus, one training recipe, one random seed, and one evaluation suite, and we swap only the tokenizer.⁴ Every result difference we report is therefore attributable to how the corpus is tokenized.

4.1 Shared model and training configuration

We train a Llama-style decoder-only Transformer (Touvron et al., 2023)⁵ from scratch under the standard autoregressive (causal) language-modeling objective. The configuration is held identical across conditions: hidden size 768, 12 layers, 12 attention heads, intermediate size 3072, a context length of 256 tokens, rotary position embeddings (Su et al., 2021), tied input and output embeddings, and attention dropout 0.1. We train for 10 epochs over the corpus. Optimization is done with Muon (Jordan et al., 2024) (learning rate 0.04) on the two-dimensional weight matrices, with an AdamW (Loshchilov and Hutter, 2019) auxiliary optimizer (learning rate 1×10^{-3} , weight decay 0.3) for the embeddings and normalization parameters. This autoregressive configuration is used for the controlled comparison in Section 5; the submitted model for the BabyLM competition (Section 5.3) adds masked-next-token-prediction (MNTP; BehnamGhader et al., 2024; Charpentier and Samuel, 2024) and vocabulary pruning (dropping morphemes seen fewer than three times in the corpus) on top.

The learning-rate schedule, batch size, epoch count, seed, and every other hyperparameter are the same in every run. The only quantity that changes across conditions is the vocabulary (and hence the number of rows in the embedding table) which is a property of the tokenizer.

4.2 Data and preprocessing

We use the 10M-word English corpus from the BabyLM 2026 `strict-small` track. We train on a cleaned version (9,732,593 whitespace-delimited words, produced by dropping CHILDES

⁴Unavoidably, this also changes the embedding matrix.

⁵Using HuggingFace’s LlamaForCausalLM

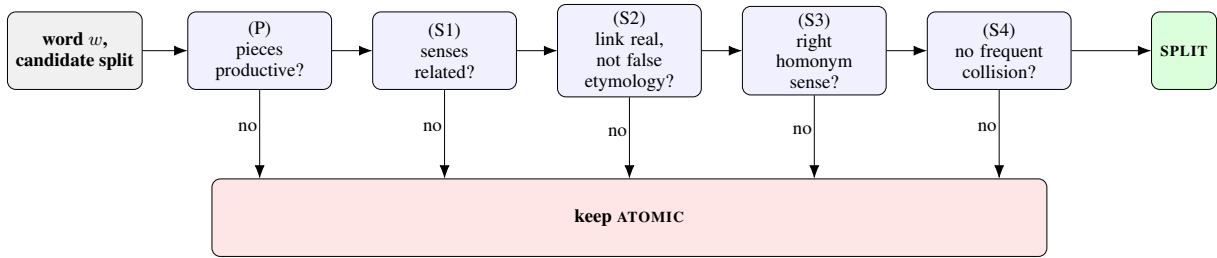


Figure 1: Decision procedure: a candidate split is accepted only if it satisfies conditions (P) and (S1)–(S4). Failing any one keeps the word atomic. Two further cases keep a word atomic in addition to these: genuine ambiguity (e.g. *leaves*) and proper names with no common-noun use (e.g. *hayward*).

transcriber comments and fixing encoding garble and OCR artifacts in OpenSubtitles and Simple Wikipedia. The corpus is lowercased and all tokenizers use whitespace pre-tokenization.

4.3 Building the tokenizer

Our tokenizer is built on MorphyNet’s morphological database (Batsuren et al., 2021), modifying and extending it as described below. The MorphyNet tables for derivational and inflectional morphology are used differently because their error rates differ sharply.

Inflectional morphology (grammatical endings: *plays, played, playing*). MorphyNet’s inflectional table is clean enough to load directly; only ~988 forms needed repair: splits that left a non-word stem (*seemed*→*seeme d*) or forms with competing analyses that needed disambiguation (*number* parsed as *numb er*). These were found with rule-based detectors and fixed manually.

Derivational morphology (word-formation: *nation*→*national*). MorphyNet’s derivational table records historical etymology, not modern structure, so it is too noisy to load directly. For example, it proposes *music*=*muse ic*, *king*=*kin ing*, *remember*=*re member*. We use it only as a source of candidates to be validated, and supplement it with independently annotated words.

As both an independent check on MorphyNet and as a source of derivational morphological splits, we hand-segmented 22,137 high-frequency words drawn primarily from LDOCE (Mayor, 2009) headwords and their definitions. This hand verification rejected ~53% of the MorphyNet derivational proposals these words overlapped with, rescuing ~572k corpus tokens from etymological over-splitting.

For the MorphyNet derivational words the hand-segmented vocabulary didn’t cover, the next most frequent 7,600 words in MorphyNet’s derivational table were audited by parallel LLM agents: a 1,000-word sample found an ~11% error rate; after fixing the flagged entries, a fresh 1,200-word sample found ~5%.

Additional words Words the BabyLM corpus contains but MorphyNet does not cover, such as run-together words (*sportclub*→*sport club*), encoding garble, typos, elisions, and approved rare words, were segmented by hand with LLM assistance.

Curation and validation The curation workflow throughout consisted of LLM agents (Claude Code) reviewing each candidate adversarially, and the author making every final decision. As an independent check of quality, we compared our segmentations against MorphoLex (Sánchez-Gutiérrez et al., 2018), an expert-built derivational database of ~70,000 English words (Appendix H): the two resources agree on ~50% of words; we under-segment relative to MorphoLex on ~38%, but these are almost entirely etymological decompositions into bound Latin/Greek roots that we deliberately keep whole; compared to MorphoLex we over-segment on only ~1%.

Tokenizer statistics In total the tokenizer records an explicit decision for 28,571 words on top of MorphyNet: ~10,000 substantively change it (1,114 splits rejected, 3,933 re-segmented, 4,961 added); the remainder confirm a MorphyNet split or keep the word whole.

4.4 Using the morphological tokenizer

To use our curated, morphological tokenizer built on MorphyNet (Batsuren et al., 2022), for pre-training, we first run it over the 10M corpus

and collect every distinct morpheme that is not an `<unk>` piece – 36,638 morphemes – and add four special tokens (`<pad>`, `<unk>`, `<bos>`, `<eos>`). Crucially, the tokenizer has *no* sub-word or BPE fallback: any piece it cannot resolve to a real morpheme is mapped to the single `<unk>` token rather than backed off to characters or byte pairs.⁶ So that the model can recover where words begin, we mark word boundaries with a fused word-initial prefix on the first morpheme of each word (like BPE’s `▮`), e.g. `unhappiness`→`▮unhappy ness`.

4.5 Baseline tokenizers

We compare against Unigram, Wordpiece, BPE, Morfessor 2.0, and MorphPiece. Unigram (Kudo, 2018), and WordPiece are each trained on the identical 10M corpus to a vocabulary of 32,000 – a standard size chosen to be close to our 36,638-morpheme inventory so that vocabulary size is not the axis of comparison. Morfessor 2.0, an unsupervised minimum description length (MDL) segmenter, is trained at its default settings and reported at whatever vocabulary it induces (68,874 here). Since MorphPiece is not open source, we reimplemented it, using the MorphyNet table with a BPE fallback and *no* curation (26,869 tokens). It shares our exact lexical source (MorphyNet), so the only difference between it and our tokenizer is our curation, stem normalization, and no-fallback design; it isolates the effect of the curation itself rather than of consulting a morphological resource at all.

4.6 Evaluation

We evaluate every trained model on the BabyLM 2026 zero-shot suite: BLiMP and the BLiMP Supplement (grammatical knowledge), EWoK (world knowledge), entity tracking, and COMPS (conceptual knowledge). All of these are scored by the standard minimal-pair method: for a pair of candidate continuations we compare the summed token log-probabilities the model assigns to each, count the pair as correct when the acceptable candidate is more probable, and macro-average accuracy per phenomenon. Because the tokenizers differ in fertility (tokens per word; Table 4), a fixed span of text becomes a different number of tokens under each, so we hold the context length constant in to-

kens rather than in words and report tokens-seen alongside the scores.

5 Results

5.1 Comparing the tokenizers on their own

We first compare the six tokenizers intrinsically on the same 10M corpus, with no model in the loop: **Ours**, BPE, Unigram, and WordPiece at a matched 32k vocabulary, Morfessor 2.0, and a reimplemented MorphPiece (raw MorphyNet plus a BPE fallback). Vocabulary size, fertility, and OOV are reported in Table 4 (Appendix A). Ours sits at a moderate fertility (1.18 tokens/word) with a 36,638-morpheme vocabulary comparable to the 32k baselines and holds OOV to 1% of tokens with no fallback, while Morfessor barely segments at its default (fertility 1.015) and raw MorphPiece covers only 19.3% of corpus tokens from its morpheme table.

Table 2 compares the tokenizers on a fixed set of words, chosen to expose where a purely statistical split diverges from a semantically-faithful one: derived words that should share a stem (*unhappiness*, *happiness*), words a statistical splitter mangles across a false boundary (*adder*, *singleton*), lexicalized wholes (*business*, *werewolf*), and transparent derivations (*nationalistic*, *disagreement*).

The contrast with the *uncurated* MorphPiece column shows that without the semantic constraint, the same MorphyNet base over-decomposes lexicalized wholes (*business*→*busyness*, *werewolf*→*were wolf*) and produces incorrect splits (*running*→*run er ing*, *happiness*→*hap y ness*), while Ours keeps those items whole and splits only where meaning is preserved.

The curation, quantified. The gap between **Ours** and **MorphPiece (raw)** columns of Table 2 is measurable hand curation: 28,571 words carry an explicit decision (19,982 split, 7,344 kept whole, 1,245 normalized). Against the *raw* MorphyNet table MorphPiece consults, our decision differs for roughly ten thousand words — we reject 1,114 splits it would make (S1–S4), change 3,933 it mis-segments (*kingdom*→*kin ing dom* becomes *king dom*), and add 4,961 it misses.

How much sharing does this buy? In total, 15.8% of corpus tokens fall in words we decompose. Of the 36,638 morphemes Ours emits, 41.9% are reused across two or more word forms; the rest are whole words kept atomic for which

⁶However, for our BabyLM submission we do use character-level fallback in order to handle OOV items in the BabyLM evaluation data.

Word	Ours	BPE	Unigram	WordPiece	Morfessor	MorphPiece (raw)
business	business	business	business	business	business	busy ness
adder	adder	ad der	adder	add ##er	adder	add er
nutrition	nutrit ion	nutrition	nutrition	nutrition	nutrition	nutrition
nutritious	nutrit ous	nutritious	nutritio u s	nutri ##ti ##ous	nutritious	nutrition ous
unhappiness	un happy ness	unhappiness	unhappi ness	unhapp ##iness	unhappiness	un hap y ness
happiness	happy ness	happiness	happi ness	happiness	happiness	hap y ness
seeming	seeme ing	seeming	seeming	seeming	seeming	seeme ing
singleton	singleton	sing leton	single t on	single ##ton	singleton	single ton
crier	cry er	crier	crie r	crier	crier	cry er
suburbia	suburb ia	subur bia	suburb i a	suburb ##ia	suburbia	suburb ia
caliphate	caliph ate	caliph ate	caliph a t e	caliph ##ate	caliphate	caliph ate
running	run ing	running	running	running	running	run er ing
nationalistic	nation al ist ic	national istic	national istic	nationalist ##ic	national istic	nation al ist ic
werewolf	werewolf	werewolf	werewolf	werewolf	werewolf	were wolf
undoing	undo ing	undoing	undo ing	undo ##ing	undoing	un do ing
disagreement	dis agree ment	disagreement	disagreement	disagreement	disagreement	dis agree ment
prehistoric	pre history ic	prehistoric	pre historic	prehistoric	prehistoric	pre history ic
tokenization	tokenization	token ization	token ization	token ##ization	token ization	token ization

Table 2: Qualitative segmentation comparison. Each cell is the token sequence for the word. WordPiece’s ## marks a word-internal continuation piece. The uncured MorphPiece column (same MorphyNet base as Ours, no semantic constraint, and with BPE fallback) over-splits lexicalized wholes and yields noisy chains where Ours does not.

no productive split exists. Among the 41.9% of morphemes that are shared, the typical stem appears in about 3 word forms (e.g., help pools signal from helped, helping, helps), giving it roughly 36% more training exposure than its most frequent word form alone.

5.2 Comparing models trained on different tokenizers

The intrinsic comparison shows the tokenizations differ. But does this difference *matter* to a language model trained with it? We pretrain one language model per tokenizer, holding everything else identical (Section 4): the same 12-layer, 768-hidden architecture, optimizer and schedule, 10M corpus, and the same seed. The only thing that varies is the tokenizer — and, unavoidably, the size of the embedding table it induces. Table 3 reports downstream accuracy on BLiMP, the BLiMP supplement, and COMPS, alongside each condition’s parameter count.⁷

Curation is what matters. The cleanest comparison in the table is Ours vs. the reimplemented MorphPiece: both draw their morphemes from the *same* MorphyNet source, and differ only in our curation and no-fallback lemma scheme. Ours reaches 66.35 BLiMP against MorphPiece’s 64.96 — a **+1.4-point gain from curation**. It is not that a tokenizer consults morphology that helps, but *how faithfully* it does so.

⁷For this comparison, the Ours condition restricts MorphyNet to its 160,000 most frequent forms (by wordfreq), yielding 36,638 morphemes; the submitted model (Section 5.3) uses MorphyNet in full.

Tokenizer	Params	BLiMP	Supp.	COMPS
Ours (no fallback)	145.0M	66.35	57.60	56.91
Ours + char. backoff	145.2M	67.13	57.21	—
BPE	146.3M	67.26	57.46	56.21
Unigram	145.3M	65.28	58.54	56.43
WordPiece	141.2M	66.20	55.45	56.09
Morfessor 2.0	175.6M	67.18	57.35	56.74
MorphPiece (raw MorphyNet)	138.9M	64.96	58.11	57.10

Table 3: Downstream accuracy under an identical model and training recipe, varying only the tokenizer. *Params* is the total parameter count (non-embedding parameters are identical across conditions; the spread is entirely the embedding table). *Supp.* is the BLiMP Supplement. All conditions use a plain autoregressive objective to isolate the tokenizer; the submitted model (Section 5.3, 72.53 BLiMP) adds MNTP and vocabulary pruning.

On COMPS (conceptual knowledge) the three morphology-based tokenizers occupy the top three positions (MorphPiece 57.10, Ours 56.91, Morfessor 56.74), with the purely statistical tokenizers trailing.

On BLiMP, BPE outperforms Ours (67.26 vs. 66.35). Ours wins slightly more BLiMP phenomena than it loses (35 of 67), and wins them where structure spans the sentence — NPI licensing (+10 to +13), Principle-A binding (+14.8), islands (+11.3), ellipsis (+9.9), irregular subject–verb agreement (+12.2). But it loses a handful of phenomena by large margins: determiner–noun agreement (−27) and superlative quantifiers (−39). Those categories, where splitting a determiner or inflection fragments a local agreement cue, drag the average below BPE.

MNTP reverses the BPE advantage. Table 3 trains under a plain autoregressive objective (with the same Muon optimizer used throughout). When we swap in the masked-next-token-prediction (MNTP) objective (BehnamGhader et al., 2024; Charpentier and Samuel, 2024) of our BabyLM submission recipe — holding the optimizer, tokenizer, and everything else fixed — and re-run the Ours-vs-BPE comparison, the ranking reverses: Ours reaches 67.71 BLiMP against BPE’s 67.11.

Finally, character backoff is worth +0.8 BLiMP over the no-fallback tokenizer (67.13 vs. 66.35) (Section 5.3); this ablation uses the 160k-restricted tokenizer, not the submission’s full-MorphoNet analyzer.

Further results The full BabyLM submission recipe tokenizer comparison across all six tokenizers is in Appendix C (Table 6). A model-size scaling curve can be found in Appendix D (Table 7).

5.3 BabyLM Submission

For the BabyLM 2026 `strict-small` (10M) track we submit a single model that pairs our tokenizer with the strongest training recipe we found. The tokenizer is the curated morphological analyzer of Section 4 with two additions: (i) a *pruned* vocabulary in which morphemes attested fewer than three times in the training corpus are dropped and routed through character backoff rather than kept as under-trained embedding rows, giving a 28,366-token vocabulary; and (ii) *character backoff* for out-of-vocabulary pieces (an unknown surface is emitted as its characters, not a single `<unk>`), so the model is never blind to a rare word — the ablation in Section 5.2 shows this backoff is worth +0.8 BLiMP over the no-fallback tokenizer. The model is a 12-layer, 768-hidden LlamaForCausalLM (~135M parameters) trained for 10 epochs with the Muon optimizer and a masked-next-token-prediction (MNTP; BehnamGhader et al., 2024) objective applied to 20% of microbatches (GPT-BERT-style hybrid; Charpentier and Samuel, 2024), with the per-token mask rate inside those microbatches decaying linearly from 0.30 to 0.15 over training and 80/10/10 corruption of the masked positions.

On the BabyLM 2026 `strict-small` leaderboard this model achieves **72.53** on BLiMP, 58.3 on the BLiMP supplement, and ranks 2nd on COMPS (54.67), 3rd on EWOK (53.2), and 7th on GlobalPIQA (38.65).

It is released as a self-contained HuggingFace repository: the analyzer, its resource tables, and the vocabulary ship with the model, and the tokenizer requires no external dependencies at load time. All tokenizer artifacts, the pretraining configuration, and the build scripts are publicly available. The character-backoff and vocabulary-pruning choices are the only submission-specific deviations from the controlled comparison of Section 5.2, which deliberately uses the plain, unpruned tokenizer with no fallback to isolate the effect of morphological segmentation itself.

Finally, training an autoregressive model with the submission tokenizer but without MNTP gives 71.29 BLiMP: **most of the ~6.2-point jump from Table 3’s 66.35 comes from the tokenizer configuration itself** (~4.9 points), and MNTP adds only ~1.2 on top.

6 Conclusion

In the low-resource regime, where every word form is thinly attested, a tokenizer that lets related forms share a morpheme is spending the data budget more efficiently, but only if the sharing is semantically related. We articulate this criterion precisely and provide a hand-curated English tokenizer that implements it. Intrinsically our tokenizer segments words into morphologically meaningful tokens, whereas statistical tokenizers cross morpheme boundaries. A comparison with an uncurated MorphPiece baseline shows that the same morphological resource, *without* the semantic constraint, over-splits lexicalized wholes, and this has downstream effects when pre-training. Our results suggest that morphological tokenization helps most when segmentation decisions are constrained by meaning, not just by boundary alignment.

Limitations

English is morphologically poor. English inflection is sparse and its derivation is heavily lexicalized, so the amount of sharing a morphological tokenizer can recover is modest. Our gains should be read as a *conservative lower bound* on what the same approach would yield for a morphologically rich language, where a single word can pack many productive morphemes and statistical tokenizers fare worst. Across Turkish (Toraman et al., 2023), Finnish and Russian (Hou et al., 2023), Hebrew (Gueta et al., 2023), Latin

(Hudspeth et al., 2026), Slovak (Držík and Kapusta, 2026), Kinyarwanda (Nzeyimana and Niyongabo Rubungo, 2022), and polysynthetic languages (Schwartz et al., 2020), morphology-aware tokenization has shown its clearest gains — and MorphyNet, our lexical backbone, already spans 15 languages, so the curated-lemma method could be extended to these languages in principle. Testing the criterion in those settings is a natural next step.

The criterion is hand-applied. The semantic half (condition S) is a human judgment; we make it explicit and identify automatable signals, but the tokenizer as delivered reflects manual curation and is not an automated pipeline. This bounds how quickly it transfers to new languages.

Maintenance. Because the tokenizer’s lookup tables are static, newly coined words, emerging senses, and shifting compositional intuitions (e.g. *ghosting* acquiring a social sense unrelated to *ghost*) would require periodic re-curation.

The gains do not extend to GLUE. They concentrate in the minimal pair evaluations and do not carry to finetuned (Super)GLUE, where our models plateau below the best statistical-tokenizer systems, and neither adaptive masking nor task-targeted synthetic data closed the gap (Appendix F).

References

- Sawsan Alqahtani, Mir Tafseer Nayeem, Md Tahmid Rahman Laskar, Tasnim Mohiuddin, and M Saiful Bari. 2026. [Stop taking tokenizers for granted: They are core design decisions in large language models](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8410–8432, Rabat, Morocco. Association for Computational Linguistics.
- Gül Sena Altintas, Malikeh Ehghaghi, Brian Lester, Fengyuan Liu, Wanru Zhao, Marco Ciccone, and Colin Raffel. 2025. [Toksuite: Measuring the impact of tokenizer choice on language model behavior](#). *CoRR*, abs/2512.20757.
- Catherine Arnett and Benjamin Bergen. 2025. [Why do language models perform worse for morphologically complex languages?](#) In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6607–6623, Abu Dhabi, UAE. Association for Computational Linguistics.
- Catherine Arnett, Marisa Hudspeth, and Brendan T. O’Connor. 2025. [Evaluating morphological alignment of tokenizers in 70 languages](#). *CoRR*, abs/2507.06378.
- Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. 2025. [Morphbpe: A morphology-aware tokenizer bridging linguistic complexity for efficient LLM training across morphologies](#). *CoRR*, abs/2502.00894.
- Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. [The SIGMORPHON 2022 shared task on morpheme segmentation](#). In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.
- Khuyagbaatar Batsuren, Gábor Bella, and Fausto Giunchiglia. 2021. [MorphyNet: a large multilingual database of derivational and inflectional morphology](#). In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 39–48, Online. Association for Computational Linguistics.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. [Llm2vec: Large language models are secretly powerful text encoders](#). *CoRR*, abs/2404.05961.
- Jean Berko. 1958. The child’s learning of english morphology. *Word*, 14(2-3):150–177.
- Necva Bölücü and Burcu Can. 2025. [A morpheme-aware child-inspired language model](#). In *Proceedings of the First BabyLM Workshop*, pages 279–287, Suzhou, China. Association for Computational Linguistics.
- Kaj Bostrom and Greg Durrett. 2020. [Byte pair encoding is suboptimal for language model pretraining](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.
- Jan A. Botha and Phil Blunsom. 2014. [Compositional morphology for word representations and language modelling](#). In *International Conference on Machine Learning*.
- Lucas Georges Gabriel Charpentier and David Samuel. 2024. [GPT or BERT: why not both?](#) In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.
- David Držík and Jozef Kapusta. 2026. [The importance of morphology-aware subword tokenization for NLP tasks in slovak language modeling](#). *Expert Syst. Appl.*, 312:131492.

- Lukas Edman and Alexander Fraser. 2025. Mask and you shall receive: Optimizing masked language modeling for pretraining BabyLMs. In *Proceedings of the First BabyLM Workshop*.
- Lennart Finke, Chandan Sreedhara, Thomas Doods, Mat Allen, Emerald Zhang, Juan Diego Rodriguez, Noa Nabeshima, Thomas Marshall, and Dan Braun. 2025. [Parameterized synthetic text generation with simplestories](#). *CoRR*, abs/2504.09184.
- 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.
- Zébulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. [From babble to words: Pre-training language models on continuous streams of phonemes](#). In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 37–53, Miami, FL, USA. Association for Computational Linguistics.
- Eylon Gueta, Omer Goldman, and Reut Tsarfaty. 2023. [Explicit morphological knowledge improves pre-training of language models for hebrew](#). *CoRR*, abs/2311.00658.
- 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.
- Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. [An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland. 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.
- 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 : Moving away from statistical language representation](#). *CoRR*, abs/2307.07262.
- Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. 2024. [Muon: An optimizer for hidden layers in neural networks](#).
- Jaap Jumelet, Michael Hanna, Marianne de Heer Kloots, Anna Langedijk, Charlotte Pouw, and Oskar van der Wal. 2023. [ChapGTP, ILLC’s attempt at raising a BabyLM: Improving data efficiency by automatic task formation](#). In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 74–85, Singapore. Association for Computational Linguistics.
- Taku Kudo. 2018. [Subword regularization: Improving neural network translation models with multiple subword candidates](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.
- Zilong Li. 2024. [Team ryu’s submission to SIGMORPHON 2024 shared task on subword tokenization](#). *CoRR*, abs/2410.17094.
- Tomasz Limisiewicz, Terra Blevins, Hila Gonen, Orevaghene Ahia, and Luke Zettlemoyer. 2024. [MYTE: Morphology-driven byte encoding for better and fairer multilingual language modeling](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15059–15076, Bangkok, Thailand. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. [CharBERT: Character-aware pre-trained language model](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 39–50, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Michael Mayor, editor. 2009. *Longman Dictionary of Contemporary English*, 5th edition. Pearson Longman, Harlow, England.
- Sherin Muckatira, Vijeta Deshpande, Vladislav Lialin, and Anna Rumshisky. 2024. [Emergent abilities in](#)

- reduced-scale generative language models. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, volume NAACL 2024 of *Findings of ACL*, pages 1242–1257. Association for Computational Linguistics.
- 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.
- Iñigo Parra. 2024. [Morphological typology in BPE subword productivity and language modeling](#). *CoRR*, abs/2410.23656.
- Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. [How good is your tokenizer? on the monolingual performance of multilingual language models](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.
- Jonne Saleva and Constantine Lignos. 2021. [The effectiveness of morphology-aware segmentation in low-resource neural machine translation](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 164–174, Online. Association for Computational Linguistics.
- Claudia H Sánchez-Gutiérrez, Hugo Mailhot, S Hélène Deacon, and Maximiliano A Wilson. 2018. MorphoLex: A derivational morphological database for 70,000 english words. *Behavior research methods*, 50(4):1568–1580.
- Lane Schwartz, Francis M. Tyers, Lori S. Levin, Christo Kirov, Patrick Littell, Chi-kiu Lo, Emily Prud’hommeaux, Hyunji Hayley Park, Kenneth Steimel, Rebecca Knowles, Jeffrey Micher, Lonny Strunk, Han Liu, Coleman Haley, Katherine J. Zhang, Robbie Jimerson, Vasilisa Andriyanets, Aldrian Obaja Muis, Naoki Otani, and 2 others. 2020. [Neural polysynthetic language modelling](#). *CoRR*, abs/2005.05477.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- 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*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.
- Jianlin Su, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. [Roformer: Enhanced transformer with rotary position embedding](#). *CoRR*, abs/2104.09864.
- Cagri Toraman, Eyup Halit Yilmaz, Furkan Sahinuc, and Oguzhan Ozcelik. 2023. [Impact of tokenization on language models: An analysis for turkish](#). *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 22(4):116:1–116:21.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [Llama: Open and efficient foundation language models](#). *CoRR*, abs/2302.13971.
- Saketh Reddy Vemula, Sandipan Dandapat, Dipti Sharma, and Parameswari Krishnamurthy. 2025. [Rethinking tokenization for rich morphology: The dominance of unigram over BPE and morphological alignment](#). In *The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 232–252, Mumbai, India. 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*, pages 1–34, Singapore. Association for Computational Linguistics.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, and 12 others. 2016. [Google’s neural machine translation system: Bridging the gap between human and machine translation](#). *CoRR*, abs/1609.08144.

A Intrinsic tokenizer comparison

Table 4 reports the intrinsic properties of the six tokenizers on the BabyLM 10M corpus (vocabulary size, fertility, and out-of-vocabulary rate), referenced from Section 5.1.

Tokenizer	Vocab	Fallback	Fertility	OOV (tok)	OOV (word)	tok/type
Ours (curated morph)	36,638	no	1.177	0.99%	1.16%	1.57
BPE	32,000	yes	1.107	0.00%	0.00%	2.37
Unigram	32,000	yes	1.329	0.00%	0.00%	2.90
WordPiece	32,000	yes	1.112	0.00%	0.00%	2.46
Morfessor 2.0	68,874	yes	1.015	0.00%	0.00%	1.64
MorphPiece (raw MorphyNet)	26,869	yes	1.287	0.00%	0.00%	2.53

Table 4: Intrinsic comparison on the BabyLM 10M `strict-small` corpus. *Fertility* is emitted tokens/word (lower = more compression). BPE/Unigram/WordPiece/Morfessor/MorphPiece carry a subword or BPE *fallback*, so they are 0% OOV *by construction*; **Ours has no fallback**, so its 1% `<unk>` tail is the price of never emitting a non-morpheme.

B Full ablation results

Beyond the controlled tokenizer comparison (Section 5.2), we ran an extensive search over training recipe and data-side interventions to build the submitted system. Table 5 collects every 10M model we evaluated. All scores are from our own reproduction of the BabyLM zero-shot scorer (BLiMP main + supplement, COMPS, EWoK, entity-tracking) and the finetuned (Super)GLUE macro-average. Dashes mark evaluations we did not run for that condition — we scored BLiMP for every model, the full suite for the core family, and (Super)GLUE for the leaderboard-relevant set.

The submitted system is **m20-ms**. “m20” denotes that base recipe (pruned morphological tokenizer + Muon + masked-next-token prediction at a 0.20 masking fraction, 135M parameters, trained on the raw 10M corpus with no data augmentation). Every “m20-*” row adds one or more data-side ingredients on top of that identical base.

C Submitted recipe tokenizer comparison

Table 3 uses a plain autoregressive objective to isolate the effects of the tokenizer. Table 6 repeats the comparison under the full submitted recipe (Muon + MNTP at 20% masking), which is the recipe behind the submitted model. **The pruned morphological tokenizer (the submission configuration) leads BLiMP by 4.4–6.5 points over every un-pruned tokenizer.** Among un-pruned tokenizers, Unigram slightly outperforms Ours-full and BPE on BLiMP, while Morfessor leads on the BLiMP supplement. Three of the five un-pruned conditions (BPE, Unigram, MorphPiece) outperform the pruned submission on GLUE.

D Effect of model size

Table 7 reports the submitted recipe (pruned morphological tokenizer) at six model sizes from 8M to 135M parameters. BLiMP improves smoothly (+2.9 from 8M to 135M), with the largest step at 14M→28M. GLUE does not improve with size and is noisy (61–65).

E Extended related work

Statistical subword tokenizers. The dominant tokenizers – BPE (Sennrich et al., 2016), WordPiece (Wu et al., 2016), and Unigram LM (Kudo, 2018) – share an objective (compress the corpus under a vocabulary budget) that is agnostic to morphology. Bostrom and Durrett (2020) show BPE is suboptimal: at a fixed vocabulary, Unigram LM matches or outperforms it downstream, while inducing segments that align more closely with morphology. Vemula et al. (2025) also find that, in their comparison of BPE and Unigram across Telugu, Hindi, and English on classification and structure prediction tasks, Unigram consistently outperforms BPE.

However, while Unigram improves statistical boundary alignment, it still fragments words across meaning, rendering e.g. `singleton` as `single t on` and `caliphate` as `caliph a t e` in our experiments, and it does not unify `-ise/-ize` or an inflectional paradigm onto one unit. Our tokenizer abandons the compression objective entirely for an explicit, semantically-constrained morphological one.

Unsupervised and resource-based morphological segmentation. Morfessor (Smit et al., 2014) induces a morphological lexicon from raw text under a minimum description length (MDL) prior. It is unsupervised and language-agnostic but optimizes for description length, not meaning, and

Model	= base submitted (m20) +	BLiMP	Supp.	COMPS	EWoK	Entity	GLUE
m20-ms (submitted)	— (submitted run)	72.53	58.44	54.66	52.98	20.31	63.9
m20-all	filter + synthetic + swap + curriculum	72.60	—	—	—	—	—
m20-super	all augmentations	72.51	58.02	56.26	52.86	—	—
m20-s3	seed 3	72.47	55.48	54.70	53.60	19.69	64.1
grand-1	filter + synthetic + swap	72.46	—	—	—	—	—
m20	— (base submitted)	72.36	58.45	54.59	52.93	20.27	63.9
m20-ns	name-swap	72.36	56.28	53.93	54.15	21.16	63.8
m20-amlm	Adaptive MLM	72.24	—	—	—	—	63.7
m10	mask-frac 0.10 (90/10)	72.18	—	—	—	—	—
m30	mask-frac 0.30 (70/30)	71.76	56.64	54.35	53.65	20.90	64.1
m15	mask-frac 0.15 (85/15)	71.57	55.73	54.15	53.42	20.11	64.9
m20-s2	seed 2	71.33	58.64	53.73	53.17	20.15	—
m20-58m	58M parameters	71.28	55.94	54.35	52.95	21.06	64.6
m25	mask-frac 0.25 (75/25)	71.15	56.19	53.88	52.85	21.41	64.8
super-58m-c512	all augment., 58M + ctx 512	70.93	57.51	56.14	53.11	22.04	—
m20-c512	context length 512	70.64	—	—	—	—	—
m20-v8nc	name-collapse	70.25	55.06	53.68	52.52	19.89	63.7
m20-cur	curriculum (alone)	69.68	56.73	53.91	52.22	20.59	63.6

Table 5: Full ablation over recipe and data-side interventions on BabyLM 10M `strict-small`. **m20** is the base recipe, with our pruned morphological tokenizer, Muon, MNTP on 20% of microbatches, 135M parameters, context length of 256, and the cleaned 10M corpus. Every other row changes one or more ingredients relative to it. **Data-side:** *filter* drops documents with >5% out-of-vocabulary tokens; *synthetic* adds Qwen-generated targeted text (Appendix F); *swap* remaps names onto evaluation-frequent names; *collapse* merges consecutive name tokens into one; *curriculum* orders the first epoch by OOV rate; *all augmentations* is all five together. **Modeling-side:** *mask-frac x (a/b)* trains MNTP on a fraction x of microbatches and causal LM on the rest; *Adaptive MLM* is from Edman and Fraser (2025); *seed 2/3* re-run the base recipe with a different random seed; *58M* and *ctx 512* change model size and context length (base: 135M, 256). **m20-ms** is a second training run of the base recipe (m20) and is the submitted model.

Tokenizer	BLiMP	Supp.	EWoK	COMPS	GLUE
Ours pruned 28k (submitted)	72.53	58.38	53.08	54.61	63.87
Ours un-pruned 41k	67.71	58.36	53.82	55.41	63.56
BPE	67.11	56.81	52.72	53.99	64.02
Unigram	67.97	56.45	50.97	54.17	64.95
MorphPiece (raw MorphyNet)	65.89	58.59	53.52	54.61	64.95
Morfessor 2.0	66.63	60.38	50.80	54.45	63.49

Table 6: Tokenizer comparison under the submitted recipe (Muon + MNTP@20%), the same recipe as the submitted model. Only the tokenizer varies.

Params	BLiMP	Supp.	EWoK	COMPS	GLUE
8M	69.43	56.42	52.16	52.79	61.43
14M	69.47	57.29	52.35	53.60	65.00
28M	71.28	56.59	53.41	53.94	63.92
58M	71.32	56.07	52.85	54.33	63.52
100M	71.81	58.82	53.08	54.11	64.22
135M	72.53	58.38	53.08	54.61	63.63

Table 7: Training the submitted recipe with various model sizes, with the pruned morphological tokenizer. All other hyperparameters held fixed. The 135M parameter model is the one we submitted to the competition.

(as we find in Section 5.1) barely segments at its default setting. Supervised morpheme segmentation, e.g. the SIGMORPHON 2022 shared task (Batsuren et al., 2022), and large morpho-

logical databases such as MorphyNet (Batsuren et al., 2021) provide gold or silver segmentations for many languages. We build on MorphyNet but do not take its segmentations as ground truth: it encodes historical and derivational structure that our criterion often rejects (Section 3), which is why the tokenizer required substantial hand curation.

Morphology-aware tokenizers for LMs. MorphPiece (Jabbar, 2023) combines a rule-based morphological table with a BPE fallback and reports gains at GPT-2 scale; it is the closest engineering precedent to our tokenizer, and we reimplement it as a baseline. FLOTA (Hofmann et al., 2022) keeps a standard vocabulary but selects the longest available segments from it, which tend to

better preserve morphological structure.

A complementary line constrains statistical training rather than replacing it: MorphBPE (Asgari et al., 2025) forbids BPE merges that cross morpheme boundaries. Byte-level schemes such as MYTE (Limisiewicz et al., 2024) apply morphology to balance sequence lengths across languages rather than to model monolingual morphology. Closest to us in setting, two BabyLM submissions train child-scale models on morphologically segmented vocabularies: Bölücü and Can (2025), using unsupervised Morfessor and a rule-based affix-stripper, and Fusco et al. (2025), who compare BPE against paradigm- and syllable-based tokenizers and conclude that no single tokenizer dominates, with morphology aiding semantics but not syntax. Unlike this work, both segment automatically and at the surface, without any semantic-relatedness constraint.

Evaluating tokenizers on morphology. A parallel line asks how well existing tokenizers respect morphology and whether it predicts model quality. Hofmann et al. (2021) show that derivationally segmented input improves generalization on rare complex words; MorphScore (Arnett and Bergen, 2025) and its 70-language extension (Arnett et al., 2025) quantify morphological alignment across tokenizers and languages; Parra (2024) relates BPE subword productivity to morphological typology across six languages. Notably, Arnett et al. (2025) find that morphological alignment explains little of the variance in downstream performance. One possible reason is that a morphologically aligned split can still be semantically misleading. Whether semantically constrained splits predict downstream gains across languages remains to be tested.

Low-resource and child-scale modeling. Several works find tokenization and vocabulary especially consequential in low-data settings: (Muckatira et al., 2024; Finke et al., 2025; Goriely et al., 2024; Li, 2024; Jumelet et al., 2023). For morphologically rich languages the stakes are higher still: morphology-aware tokenization has been reported to help across Finnish and Russian (Hou et al., 2023), Hebrew (Gueta et al., 2023), Latin (Hudspeth et al., 2026), Slovak (Držík and Kapusta, 2026), Kinyarwanda (Nzeyimana and Niyongabo Rubungo, 2022), and polysynthetic languages (Schwartz et al., 2020), though some of the evidence is mixed (Saleva and Lignos, 2021;

Toraman et al., 2023; Vemula et al., 2025), and several reported gains are in perplexity or convergence rather than downstream accuracy.

F Other interventions

Most interventions that are intuitive or that helped in other settings did not improve our system; one did. We report them because negative results are useful to the community.

Name remapping is neutral. We remapped names in the training data onto the names that recur in the evaluation suites, consistently within each training sequence, and only within gender and person/place pools,⁸ expecting better performance since the model is not wasting capacity learning a plethora of unnecessary names. It was neutral on the headline metrics: identical BLiMP (72.36), unchanged (Super)GLUE (63.8 vs. 63.9), better on EWoK/entity (+1.2/+0.9) but worse on the BLiMP supplement (−2.2).

Curriculum does not help. We tried a curriculum which orders the first epoch’s documents by OOV tier (fraction of <UNK> tokens under the pruned vocabulary, bucketed into four tiers from 0% to 5%, easy-to-hard); it also drops all documents with >5% OOV, removing 3.5% of documents (7.3% of tokens) relative to the full-corpus baseline.

The combined effect costs −2.7 BLiMP (72.36→69.68), though this conflates reordering with the data reduction. Comparing two runs that both drop the same documents and differ only in ordering, reordering alone is approximately neutral (grand-1 72.46 → m20-all 72.60, +0.14). Curriculum’s harm is offset once synthetic data is added (m20-all recovers to 72.60), but it never helps, consistent with prior BabyLM findings that curriculum learning rarely pays off (Warstadt et al., 2023).

Longer context hurts the strong recipe. Raising the context length from 256 to 512 cost

⁸Names were bucketed into four pools (male, female, gender-neutral person, place) and a name was only ever replaced by a name from the same pool, so anaphor agreement (*John...himself*) and person/place distinctions were never corrupted. The mapping was drawn fresh for each training sequence and held fixed within it, so repeated mentions of a name in one sequence stayed consistent while the same corpus name landed on different targets elsewhere. Ambiguous word-names such as *bill* or *frank* could be targets but were never replaced, so their common-noun uses were left intact.

−1.7 BLiMP (72.36→70.64) under the submitted recipe.

Multi-stream input fusion improves perplexity but not grammar. Fusing character-, morpheme-, and word-level embeddings at the input (Botha and Blunsom, 2014; Ma et al., 2020) produced the lowest validation losses of any run (down to 3.29) yet BLiMP at or below baseline (67.8–68.8). This is consistent with Edman and Fraser (2025), who found that adding sub-token embeddings lowered BLiMP accuracy by 2 to 3 points while leaving (Super)GLUE fine-tuning performance unchanged.

Synthetic data is the one data-side win. The one data-side intervention that helped was adding targeted synthetic text. We used Qwen2.5-7B-Instruct (a ≤9B teacher, which the challenge FAQ permits) to generate short, simple, child-level sentences aimed at the base model’s measured weaknesses. A first tranche (~354k words) targeted the BLiMP phenomena on which the base model erred most, with each phenomenon’s share proportional to its error rate (NPI licensing 52%, filler–gap 13%, determiner–noun agreement 11%, quantifiers 10%, reflexives 7%, irregular participles 7%); a second tranche (~458k words) exercised property inheritance (for COMPS), object–location tracking (for entity tracking), and everyday physical and social knowledge (for EWoK). The synthetic text was interleaved into the filtered corpus. Every run that included it matched or exceeded the base model on BLiMP (grand-1 72.46, m20-super 72.51, m20-all 72.60 vs. 72.36), whereas the surface-normalization interventions (name remapping, name collapse) did not. The submitted model does not use synthetic data (Appendix G).

G Training data and budget compliance

Training corpus. We train exclusively on a *cleaned* version of the official BabyLM 2026 STRICT-SMALL corpus; no other text corpus is used. Cleaning is deterministic and reproducible from the released cleaning script: the raw corpus (10,000,019 words by `wc -w`) reduces to **9,732,593** words (−2.7%). The removed material is transcription and formatting junk rather than child-directed speech: 40,073 CHILDES transcriber-comment lines (the dominant removal), OpenSubtitles mojibake/OCR ar-

tifacts, subtitle markup and watermark lines (~2,500 lines), and Simple English Wikipedia HTML entities. Post-cleaning line counts per source: BNC spoken 65,220; CHILDES 491,232; Gutenberg 59,934; OpenSubtitles 385,975; Simple English Wikipedia 58,709; Switchboard 2,892.

Word-budget accounting. Training text is 9,732,593 words (<10M); ten epochs give ~97.3M cumulative words seen (under the 100M exposure cap). The submitted model uses no synthetic data, no external model, and no distillation. The 19 required intermediate checkpoints (every 1M words seen to 10M, every 10M to 100M) are published as revisions of the model repository.

Tokenizer resources. The morphological analyzer is a lookup/rule system; no component of it is trained on external text. It consults (i) hand-curated morpheme tables created by the authors (28,544 entries) and (ii) the MorphyNet English inflectional/derivational lexicon, used in full (the analyzer tables contain 187,132 form→segmentation entries and a 617,042-item known-word list of single-word spellings — every lemma/form spelling in MorphyNet, consulted for known-word decisions during segmentation). All corpus statistics used in vocabulary construction (frequency pruning, character backoff) derive from the training corpus itself. During training the analyzer can only consult entries for words occurring in the corpus (135,085 types); this effective slice is 59,285 known-word entries, 16,968 inflection entries, and 21,535 curated entries — at most 168,757 words counted maximally strictly, so training text plus consulted lexicon annotations total 9,901,350 words, under the 10M budget even on that conservative accounting.

H MorphoLex comparison

As an independent quality check, we compared our tokenizer’s segmentations against MorphoLex (Sánchez-Gutiérrez et al., 2018), an expert-built derivational morphological database of ~70,000 English words from the English Lexicon Project.⁹ MorphoLex records only derivation on lemmas (no inflection), so we strip the inflectional ending from our output before comparing derivational structure. Table 8 shows the full breakdown.

⁹<https://github.com/hugomailhot/MorphoLex-en>

Bucket	Words	%	Definition
Identical	17,497	25.5	Both produce the same pieces
Agree on derivation	16,743	24.4	Same derivational split; we additionally peel off the inflectional suffix that MorphoLex does not represent
They split more	25,937	37.8	MorphoLex decomposes into bound Latin/Greek roots (<i>abduct</i> → <i>ab duct</i>) we deliberately keep whole
We split more	686	1.0	We over-segment relative to MorphoLex (includes some by-design cases such as suppletives)
Convention diff	2,538	3.7	Same piece count; one piece has a different canonical spelling (e.g., MorphoLex <i>accuse ion</i> vs. ours <i>accuse ation</i> ; <i>im legal</i> vs. <i>in legal</i>)
We OOV	5,216	7.6	Our tokenizer cannot analyze the word; MorphoLex can (mostly proper names and rare tokens outside our lookup tables)
Total	68,617	100	

Table 8: Comparison of our segmentations against MorphoLex on all 68,617 words in MorphoLex (after deduplication and dropping 6 data-error rows). The 68,617 is the full MorphoLex database, not filtered to any corpus.

The comparison shows that where the two resources disagree, it is overwhelmingly by design: the 37.8% “they split more” bucket is almost entirely MorphoLex’s etymological decomposition into bound roots (*abase*→*a base*, roots like *brevi*, *duct*, *plic*) that our synchronic criterion rejects for the same reason we do not split *remember*→*re member*. The comparison also confirmed that our productive-affix coverage matches MorphoLex, with zero missed *-ness/-ment/-ful/-less/-able* splits.

I Per-phenomenon acquisition curves

Figure 2 plots the per-phenomenon BLiMP accuracy of the submitted model (*morpheus-10M-v5*) across 19 official intermediate checkpoints (1M to 100M cumulative words seen, evaluated with the official *fast_eval* protocol). The phenomena separate into distinct acquisition classes:

- **Early risers (mastered by ~3–5M words):** plain determiner–noun agreement (95+ by 3M words); the irregular and adjective-intervening variants reach 80–90 by 5M and continue to climb.
- **Slow climbers:** anaphor number agreement reaches 95+; anaphor gender agreement (~70 at 5M → 96) and subject–verb agreement (65–76 at 5M → 82–88), together with passives, show steady growth through 100M words.
- **Late bloomers (emerge only after ~30–50M words, i.e. epochs 4+):** *wh* vs. *that* with gap (20→55), *left branch island simple question* (20→65), *coordinate structure constraint complex left branch* (25→52)— the long-distance

dependency family, which requires many exposures.

- **Never learned (flat or at chance through 100M words):** matrix-question NPI licensing (~10), NPI-present (~20–40), *only*-NPI licensing (wild oscillation), complex-NP island (~40), *tough-vs-raising-1* (~30), Principle-A reconstruction (~35). These are predominantly NPI licensing and hard island phenomena requiring abstract scope/licensing relations that 10M words and 135M parameters never acquire.

morpheus-10M-v5 — BLiMP per-phenomenon learning curves (official fast_eval over 19 checkpoint revisions; x = words seen, M, log)

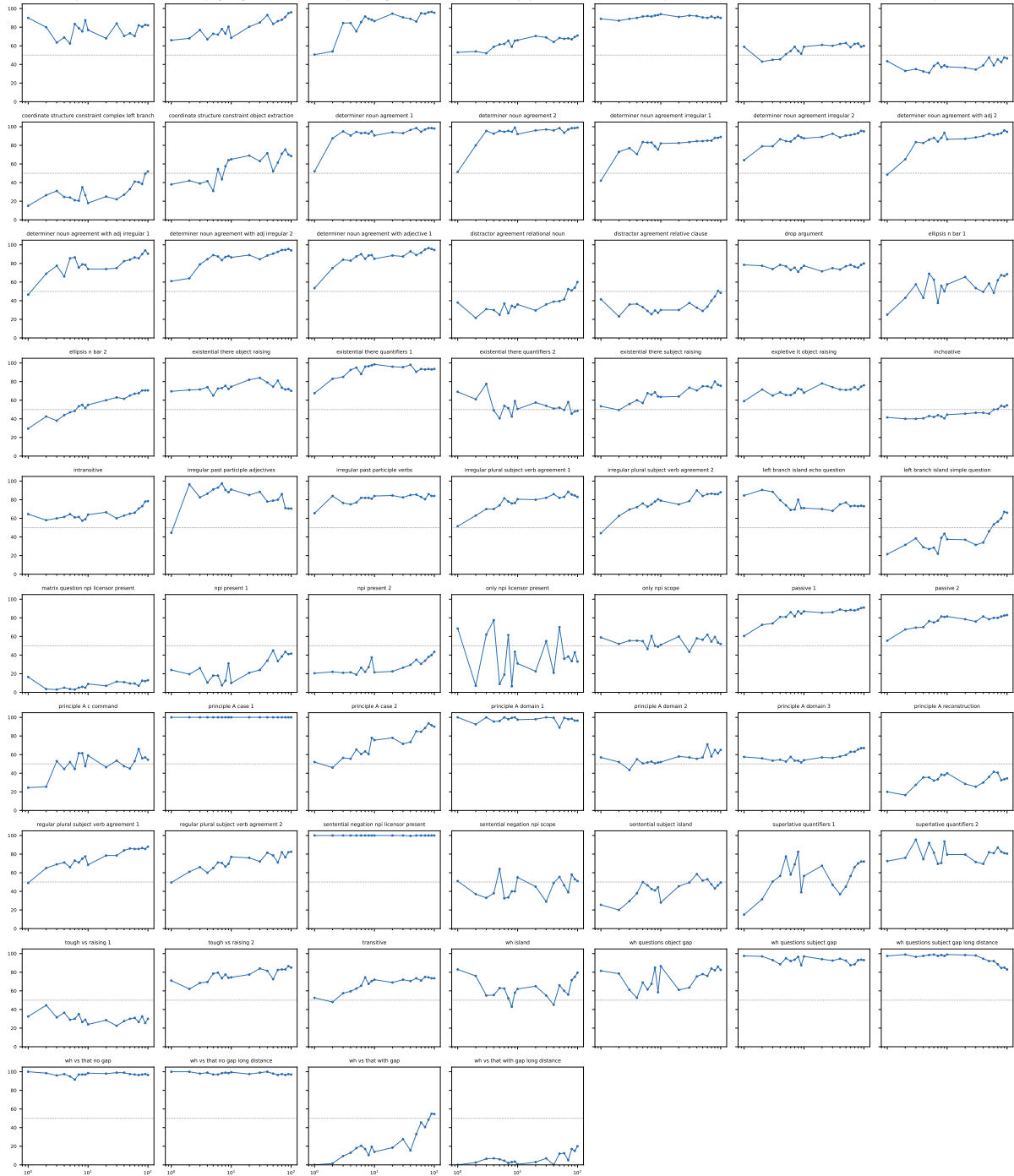


Figure 2: Per-phenomenon BLiMP learning curves for the submitted model across 19 official checkpoints (x-axis: cumulative words seen, log scale). Each panel is one BLiMP phenomenon. Phenomena separate into early risers (determiner–noun agreement, mastered by 3–5M words), slow climbers, late bloomers (long-distance dependencies, emerging after 30–50M words), and never-learned (NPI licensing, flat at chance).