Linguistic Units as Tokens: Intrinsic and Extrinsic Evaluation with BabyLM

Achille Fusco[1,2] **Maria Letizia Piccini Bianchessi**[2]
Tommaso Sgrizzi[2] **Asya Zanollo**[2] **Cristiano Chesi**[2]
[1]University of Florence, Via della Pergola 60, Florence, Italy
[2]IUSS Pavia, Piazza della Vittoria 15, Pavia, Italy
achille.fusco@iusspavia.it letizia.piccinibianchessi@iusspavia.it
tommaso.sgrizzi@iusspavia.it asya.zanollo@iusspavia.it
cristiano.chesi@iusspavia.it

Abstract

Tokenization is often treated as a preprocessing step, yet in data-limited settings it directly shapes what a model can learn. We compare four segmentation strategies in the BabyLM Challenge: merge-based BPE, morphology-aware, split-based MorPiece and ParadigmFinder, and syllable-based SylliTok. Evaluation combines two perspectives. First, an intrinsic test on the SIGMORPHON 2022 segmentation benchmark, adapted to English, measures how closely each tokenizer aligns with morpheme boundaries. Second, extrinsic tests train GPT-2 on the 10M BabyLM corpus and evaluate it on the 2025 benchmark. No single tokenizer dominates. BPE remains strong on syntax-heavy tasks, ParadigmFinder excels in semantic composition and age-of-acquisition alignment, and SylliTok shows advantages in discourse tracking. Morphology-aware tokenizers achieve the best intrinsic segmentation scores, and these gains translate into more robust generalisation in comprehension tasks. These results highlight tokenization as a core modeling decision, with direct consequences for compression, morphology, and the path to human-like learning.

1 Introduction

The BabyLM Challenge (Warstadt et al., 2023a,b) was designed to evaluate how language models acquire linguistic competence under data conditions that approximate human language learning. By restricting training to corpora of 10M or 100M tokens, the benchmark provides a testbed for exploring which modeling choices enable robust acquisition from limited input. While most submissions have focused on architecture and training objectives, a less visible but equally fundamental choice concerns the unit of tokenization. The segmentation of raw text into model input units determines not only how words are represented, but also what kinds of generalisations the model is in principle able to make.

Standard approaches such as byte pair encoding (BPE; Gage 1994; Sennrich et al. 2016) treat tokenization as a purely statistical compression problem, merging frequent character pairs without regard for linguistic structure. Recent work, however, has argued that tokenization should be viewed as an integral part of the modeling effort (Goldman et al., 2024; Oh and Schuler, 2025), shaping both the inductive biases of the system and its ability to align with humanlike generalisation. In particular, morphology has long been seen as a critical domain for testing theories of language acquisition (Goldsmith, 2001; Xu et al., 2018), and offers a natural arena for designing tokenizers that attempt to capture linguistically meaningful units.

In this paper, we ask how different linguistically oriented tokenizers affect learning in the BabyLM setting. We consider four segmentation strategies, ranging from merge-based (BPE) to split-based (MorPiece, ParadigmFinder) and syllable-based (SylliTok). To evaluate them, we combine two complementary perspectives: an *intrinsic* assessment using the SIGMORPHON 2022 morphological segmentation benchmark, and an *extrinsic* evaluation using the BabyLM 2025 test suite. This dual approach allows us to measure both how well each tokenizer approximates humanlike segmentation and how these choices influence downstream learning and generalisation.

Our results show that no single tokenizer dominates across tasks. Instead, each segmentation strategy introduces its own strengths and weaknesses: morphology-aware tokenizers excel in capturing systematic structure and supporting semantic generalisation, syllable-based segmentation contributes to discourse sensitivity, and frequency-driven BPE remains a strong all-around baseline. Taken together, these findings highlight tokenization as a substantive modeling decision, with implications for compression, morphological generalisation, and

Proceedings of the First BabyLM Workshop, pages 496–507
November 8, 2025 ©2025 Association for Computational Linguistics

the design of cognitively plausible learning systems.

2 Motivation

Tokenization is a crucial step in natural language processing, especially in the development and training of language models. It determines the basic units with which models will operate, ultimately shaping their ability to generalize, compress, and understand linguistic structure. While often treated as a technical detail, tokenization in fact sits at the intersection of practical engineering choices, information-theoretic principles, and linguistic theory. In this section, we articulate three perspectives on tokenization: as a modeling choice, as a compression strategy, and as a proxy for morphological segmentation.

2.1 Tokenization as Modeling and Compression

Tokenization is often viewed as a simple preprocessing step—particularly in languages without explicit word boundaries (e.g., Chinese, Japanese, Thai)—but in practice it defines the basic units on which a model learns. Decisions about how to segment text into words or subwords affect the handling of out-of-vocabulary items, the effective sequence length, and the allocation of parameters in the embedding layer. This introduces asymmetries across languages: scripts, morphology and resource availability lead to different degrees of token fragmentation and vocabulary inflation, which in turn influence the performance of language models trained on comparable amounts of data.

A second and equally important aspect is that tokenization originates in data compression. Byte Pair Encoding (BPE), now ubiquitous in NLP, was first introduced by Gage (1994) as a general-purpose compression method. The algorithm iteratively replaces the most frequent pair of adjacent bytes with a new symbol and stores the mapping in a lookup table; modern tokenizers adopt the same strategy but stop after a fixed number of merges to obtain a desired vocabulary size. More recently, Goldman et al. (2024) show that the compression capacity of a tokenizer correlates strongly with downstream model performance: tokenizers that reduce entropy more effectively tend to yield better models, especially in low-resource settings. Seen through this lens, tokenization is not an afterthought but an intrinsic modeling decision that balances representation granularity, computational efficiency and information retention. Our experiments therefore compare not only frequency-based BPE but also linguistically informed alternatives, asking how different choices of basic units affect both compression and learning.

2.2 Tokenization as Morphological Segmentation

At the same time, tokenization is intimately related to the linguistic structure of words. Natural languages are compositional at multiple levels, and morphology provides some of the clearest evidence of this: words are built from smaller, meaningful units — morphemes — such as roots, inflectional markers, and derivational affixes. An ideal tokenization system would capture these junctures, segmenting text in a way that reflects its internal linguistic organization.

For example, while a frequency-based tokenizer might store *dog* and *dogs* as separate units, a morphologically-aware tokenizer would recognize that the plural form is derived from the singular by appending a regular inflectional morpheme *-s*. Segmenting at this level reveals productive patterns that aid generalization, allowing the model to infer the meaning and form of novel or rare words from their components.

Morphological segmentation has long been studied as a core component of linguistic theory and cognitive modeling. The psychological reality of morphemes is attested in experiments like the WUG test (Berko, 1958), where infants reliably extend morphological rules to novel forms.

These experiments indicate that children exhibit clear sensitivity to the distributional properties of morphemes in their linguistic input, an ability typically classified under statistical learning (Sandoval et al., 2017; Mehler et al., 1988). As learners internalize these patterns, they begin to abstract and generalize morphological rules following distinct trajectories, a phenomenon evidenced by systematic overgeneralization errors like *goed* for *went*, or *falled* for *fell* (Lignos and Yang, 2016). Indeed, the acquisition of morphological rules appears to follow what is known as the *Tolerance–Sufficiency Principle* (Yang, 2016), which provides a formal account of when a linguistic rule can be considered productive given the sparsity and irregularity of the input data. More concretely, the Tolerance Principle (TP) states that if a rule R may potentially apply to a set of N items, then R is productive if

and only if the number of exceptions e satisfies (1):

$$e \leq \theta_N \text{ where } \theta_N = \frac{N}{\ln N} \quad (1)$$

When the number of exceptions exceeds this threshold ($e > \theta_N$), the learner is expected to treat those cases as lexicalized, and the rule R is considered unproductive. A complementary formulation is provided by the *Sufficiency Principle* (SP), which specifies the minimum amount of evidence required to support an observed generalization. Formally, given a generalization R over N items, where R is attested in M cases, R is extended to the remaining $N - M$ items if and only if:

$$N - M \leq \theta_N \text{ where } \theta_N = \frac{N}{\ln N} \quad (2)$$

The question, then, is whether artificial systems can similarly benefit from identifying morphemes — and whether doing so would support better compression, generalization, and interpretability. Subword tokenization (e.g., Byte-Pair Encoding) is in fact able to capture recurring internal structure within words, such as prefixes, suffixes, and roots (e.g., *un-* + *believ* + *able*), allowing the model to generalize across unseen words. This can be viewed as a form of compositional representation that mirrors the generative flexibility of human morphology. However, unlike morphemes, LLM tokens are not guaranteed to be semantically meaningful, and their segmentation is only driven by frequency optimization rather than grammaticality or communicative function. A truly cognitively plausible tokenizer is unlikely to achieve optimal efficiency as defined purely by compression or predictive performance. Instead, it might display the kinds of overgeneralization errors and irregularities that characterize child language acquisition. Such "imperfect" segmentation reflects the underlying learning process rather than a finished, fully optimized system.

2.3 Toward an Integrated Perspective

The three perspectives above—tokenization as modelling, compression and morphological segmentation—are complementary rather than competing. They suggest that linguistic plausibility, information-theoretic efficiency and engineering convenience can, in principle, be aligned. To put this hypothesis to the test, we explore a diverse family of tokenizers designed to identify basic linguistic units (syllables and morphemes) as tokens.

We evaluate these tokenizers along two complementary axes. First, we assess each tokenizer on its own by measuring how well it segments words into morphemes using a re-adapted version of the SIGMORPHON 2022 morpheme-segmentation benchmark (Batsuren et al., 2022), quantifying their morphological "soundness." Second, we pair each tokenizer with a fixed GPT-2 architecture and train on the BabyLM 2025 strict-small corpus. The resulting models are evaluated on the BabyLM challenge suite of tasks—ranging from linguistic preference tests (BLiMP, BLiMP-Supplement, EWoK) to downstream fine-tuning (GLUE)—as described by Warstadt et al. (2023b). By correlating segmentation quality and model performance, we aim to clarify whether linguistically motivated tokenizations lead to tangible benefits for small-scale language modelling.

The next sections build on this motivation. Section 3 surveys prior work on unsupervised morphology, paradigm discovery and the role of tokenization in language modelling, providing the theoretical context for our tokenizer designs. Section 4 details the datasets, tokenizer construction and evaluation used in our experiments, followed by an analysis of results across both morphological and BabyLM benchmarks.

3 Related Work

3.1 Unsupervised morphological segmentation

Morphological segmentation has long been viewed as both a descriptive and an information-theoretic problem. Goldsmith (2001) introduced the *Linguistica* system, framing the discovery of morphemes and paradigms as a minimum description length (MDL) optimization task. By balancing model complexity against data fit, the algorithm learns a lexicon and a set of affix patterns that jointly minimize description length, using these patterns to predict segmentation points in unseen words. Subsequent work by Xu et al. (2018) proposed a probabilistic model that identifies roots, suffixes, and transformation rules to generate candidate segmentations for each word, and then induces shared paradigms to filter out spurious affixes. Both studies demonstrate that paradigm extraction is critical for capturing the combinatorial nature of morphology; this insight motivates our morphology-oriented tokenizers, which aim to discover recurring patterns rather than simply splitting words into arbitrary subword units.

3.2 Tokenization and compression in language modelling

Subword methods such as byte-pair encoding (BPE) are now ubiquitous, yet recent research has emphasized the deeper role tokenization plays in language modelling itself. Goldman et al. (2024) systematically investigate how different tokenizers affect model performance through the lens of text compression, showing that tokenizers with lower empirical entropy (i.e., greater compression) tend to yield better downstream performance. They argue that tokenization should be viewed as an integral component of the modelling pipeline rather than as a mere preprocessing step. Fusco et al. (2024) first proposed MorPiece, a tokenization strategy based on a Trie structure for the representation of the entire lexicon, identifying splits though the application of the Tolerance–Sufficiency Principle (Yang, 2016) (see also Section 4.2). Oh and Schuler (2025) examine how token granularity influences the predictive power of language models' surprisal measures relative to human processing data. Bunzeck et al. (2025) compare grapheme- and phoneme-based small models, finding that they perform comparably to their subword analogues trained on the same limited token budget. Pagnoni et al. (2025) introduce an LLM architecture that eliminates tokenization altogether, instead representing text as variable-length byte patches defined dynamically by entropy. Finally, Raj S et al. (2025) employ a Viterbi-like algorithm that also operates on a Trie-based representation of the vocabulary to compute globally optimal segmentations, reporting improvements over the greedy BPE baseline on both intrinsic and extrinsic metrics.

3.3 Morphological segmentation as a shared task

The SIGMORPHON series of shared tasks provides a valuable benchmark for evaluating morphological models. In the 2022 edition of the morpheme segmentation task, Batsuren et al. (2022) challenged systems to decompose words and sentences into sequences of morphemes across a diverse set of languages. Subtask 1 comprised a 5 million-word corpus covering nine languages, while subtask 2 involved sentence-level segmentation in three languages. The best systems achieved an average F_1 score of 97.3 % across languages and outperformed standard tokenizers such as BPE and Morfessor by more than 30 percentage points. These results demonstrate that data-driven morphological segmenters can capture complex derivational and inflectional patterns and that morphological segmentation yields more accurate boundaries than generic subword methods. We adapt the SIGMORPHON 2022 data for evaluating our tokenizers, using its gold-standard segmentations to quantify how well each tokenizer preserves morphological structure.

3.4 Implications for BabyLM and our study

Previous BabyLM submissions highlight the importance of model architecture and input representation. The GPT-BERT model of Charpentier and Samuel (2024) shows that combining masked and causal objectives can improve BabyLM scores by enabling a single transformer to operate in both modes. The phoneme-based approach of Goriely et al. (2024) demonstrates that non-standard tokenizations (phonemic transcription, character-level segmentation) can yield competitive performance, albeit with trade-offs such as slight drops on text-based tasks. Our work continues this tradition by systematically comparing GPT-2 models trained with multiple linguistically oriented tokenizers—including BPE, character-level, morphology-based and hybrid variants—and relating their BabyLM performance to their ability to segment words morphologically. By bridging insights from unsupervised morphology, tokenization compression theory and SIGMORPHON evaluations, we aim to better understand how tokenization choices shape the learning and generalisation of small language models.

4 Experiments

In this section we describe the data, tokenizers, model configurations and evaluation protocols used in our study. Wherever possible, we follow the BabyLM challenge guidelines to ensure comparability with prior work.

4.1 Data and Preprocessing

BabyLM corpus. Our training data come from the BabyLM 2025 strict-small track, which offers a fixed corpus of roughly 10M words. The dataset comprises text from six distinct domains that reflect diverse linguistic contexts. Data are taken from conversational sources, with CHILDES contributing 29% child-directed dialogue data and OpenSubtitles providing 20% scripted dialogue, while written materials include Project Gutenberg's fiction

and nonfiction works (26%) and Simple English Wikipedia entries (15%). Additional dialogue data comes from the British National Corpus (8%) and Switchboard telephone conversations (1%). We applied a lightweight preprocessing, consisting of space normalization, lowercasing and separation of alphabetic characters from digits and punctuation, except for apostrophes.

SIGMORPHON segmentation benchmark. In order to perform an *intrinsic* evaluation of our tokenizers, independently of the language model, we turn to the word-level morpheme segmentation dataset released as part of the SIGMORPHON 2022 shared task on morpheme segmentation. The organisers provided gold segmentations for nine languages (Czech, English, Spanish, Hungarian, French, Italian, Russian, Latin and Mongolian) and reported that the best systems achieved an average F_1 score of 97.3 % across languages. Importantly, the task distinguishes between two kinds of segmentation. The original "deep" or *canonical* segmentation aligns each segment with an underlying lemma; morphemes are restored to their canonical shapes even if surface forms have undergone phonological or orthographic changes. For example, the English noun *collision* is canonically segmented as *collide+ion*, rather than its surface segmentation *collis+ion*; likewise, *profitably* would be segmented as *profit+able+-ly*. This canonical segmentation is "deeper" in that it recovers latent morphological structure beyond simple boundary detection, effectively lemmatising each morpheme.

In this work we focus on a more practical, "shallow" segmentation that is closer to tokenization. We convert the canonical segmentations provided in the SIGMORPHON test set into surface-level boundaries by simply inserting split markers at morpheme boundaries without altering the character sequence. That is, we segment *collision* as *collis+ion* and *profitably* as *profit+ably*, leaving the surface text unchanged. This conversion (i) avoids introducing extra graphemes or lemma forms that would not appear during training, (ii) aligns the task with tokenization, where the goal is to identify basic units in the input rather than to normalise them, and (iii) is theoretically motivated by the view that morphological composition operates over both roots and affixes, so a tokenizer should aim to segment wherever composition occurs—even if the base form shows no internal change. The difference between deep and shallow segmentation is less pronounced in English than in languages with richer fusional morphology, but the shallow version still provides a useful proxy for measuring how well a tokenizer captures morpheme boundaries. In our experiments we extract only the English portion of the SIGMORPHON test set and use the resulting shallow segmentation as a gold standard for evaluating each tokenizer's morphological soundness (Section 4.4).

4.2 Tokenizers

For our experiments we compare the standard byte-pair encoding (BPE) tokenizer (Gage, 1994; Sennrich et al., 2016) against three linguistically motivated tokenizers: *MorPiece* (MoP) (Fusco et al., 2024), *SylliTok*, and *ParadigmFinder* (ParFind). All tokenizers are trained on the BabyLM 10M training data with a maximum vocabulary size of 30 000 tokens, using identical preprocessing. Below we summarise the key design principles of each.

MorPiece (MoP). MorPiece segments words into morpheme-like units by postulating a split whenever the Tolerance–Sufficiency Principle (SP) (Yang, 2016) can be applied during the traversal of the lexicon. The current implementation diverges minimally from the original MoP model (Fusco et al., 2024) while preserving its Trie-based lexical structure. Consider the word $cats$: a root Trie ($c \to a \to t \to s$) and an inflectional Trie ($s \to t \to a \to c$) are created. "Traversing" the lexicon entails incrementing by one the counter of each node encountered in both the root and inflectional tries. If a path does not exist, it is assigned an initial value (i.e., frequency) of one. A split between t and s is postulated if, and only if, the SP is satisfied in both the root Trie and the inflectional Trie, that is:

$$\text{split } \textit{iff} \text{ in } \textit{root-trie}: \quad \frac{\text{freq}(t)}{\ln(\text{freq}(t))} > \text{freq}(s) \qquad \text{and in } \textit{infl-trie}: \quad \frac{\text{freq}(s)}{\ln(\text{freq}(s))} > \text{freq}(t)$$

If this is the case, the s pendant (in this instance, simply s) is added to the root Trie, rather than to the special root node "++" as in the original MoP. At the end of processing, all nodes with a frequency below the *min_freq* parameter are pruned. A MaxLength strategy is then applied to retrieve the tokens for each word during encoding.

In our experiments, the training procedure was constrained to prune the dictionary according to the *min_freq* parameter and to save a vocabulary version every 100K tokens of exposure. The resulting

vocabularies are useful for verifying each splitting hypothesis and the evidence required to postulate it (with the order of acquisition, *ooa* parameter, set to *True*).

During vocabulary construction, the algorithm traverses the Trie and identifies candidate segmentation points according to two hyper-parameters: a *cutoff* threshold and a *branching factor* (*bf*). The *cutoff* specifies the minimum frequency a mother node must reach before a split is postulated (set to 100 in all our experiments). The *branching factor* specifies the minimum number of distinct daughters a mother node must have in order for the SP to apply (set to 2 in all our experiments).

Unlike BPE, MorPiece relies on linguistic cues—high type frequency and morphological variability—to determine split points. It does not depend on precompiled morpheme lists (as in (Jabbar, 2023)), but instead induces potential morphemes directly from the data using the trie. The settings we adopted favor plausible segmentations and capture frequent inflectional and derivational affixes (e.g., *-ed*, *-ing*, *-s*, *-ness*, *un-*) while preventing oversegmentation of rare strings. However, the current vocabulary-building procedure does not delete a pendant when a split is postulated; pendants are removed only during the pruning step of the optimization phase. For this reason, we applied an aggressive optimization strategy, pruning nodes below the *min_freq* threshold every 100K tokens of exposure. This process yielded a vocabulary of approximately 23K tokens under the *strict-small* training regime and about 40K tokens under the *strict* training regime.

SylliTok. SylliTok is a rule-based tokenizer designed to align token boundaries with the syllabic structure of English. Linguistic and psycholinguistic research has shown that infants are highly sensitive to syllable-level patterns in continuous speech, often segmenting syllables before larger morphological units. Building on this insight, SylliTok uses deterministic syllabification rules to split words into syllables, yielding a token vocabulary of size 20K. For example, *banana* is tokenised as *ba-na-na* and *computer* as *com-pu-ter*. In languages with relatively transparent orthography, such as Spanish and Italian, the mapping from orthography to syllables is straightforward; in English it is more complex due to inconsistent spelling–sound correspondence. Nonetheless, a syllable-based tokenizer provides a cognitively plausible baseline and reduces token length in a way that may benefit low-resource models.

ParadigmFinder (ParFind). ParFind is another unsupervised tokenizer that extracts paradigms from the vocabulary and uses them to segment words, following previous work by Goldsmith (2001) and Xu et al. (2018). In our framework, a *paradigm* consists of a set of roots and a corresponding set of suffixes that co-occur with systematic regularity. For example, the words *walk*, *walks*, *walked* and *walking* are evidence for a paradigm with root *walk* and suffixes {`-∅`, `-s`, `-ed`, `-ing`}. ParFind induces such paradigms from the data in a multi-step process. The search is initialized by enumerating all possible binary splits of words into candidate roots and suffixes, and then grouping together roots that share identical suffix sets (see Fig. 1). The algorithm then normalizes suffixes by factoring out common prefixes and appending them to the roots, ensuring that segmentation points correspond to genuine morphological variation in at least one case in each paradigm. This step prevents the formation of spurious paradigms such as {`-t`, `-ts`, `-ted`, `-ting`}, which arise when several verb roots share the final letter (e.g., *-t*) (see also Goldsmith, 2001 on this issue). At this point, paradigms are expanded according to a Tolerance–Sufficiency Principle, enabling generalization to unseen forms. Formally, given two paradigms P_i and P_j with root sets R_i and R_j and corresponding suffix sets S_i and S_j, where $|S_i| < |S_j|$, P_i is merged into P_j if and only if a majority of roots in R_j occur in R_i, that is, if

$$|R_j| - |R_i \cap R_j| \geq \theta_{|R_j|} \quad \text{where} \quad \theta_{|R_j|} = \frac{|R_j|}{\ln |R_j|} \tag{3}$$

This condition ensures that paradigms are merged only when the overlap between root sets provides sufficient evidence for systematic extension rather than accidental co-occurrence. When words can still be analyzed with different segmentations according to multiple paradigms, suffixes are checked against existing paradigms to determine whether nested suffixation is possible. For instance, the words *singer* and *singers* can be segmented as *singer-∅* and *singer-s* under paradigm P_1, and as *sing-er* and *sing-ers* under paradigm P_2. In this case, since P_1 is already a productive paradigm, *-er* and *-ers* are in turn analyzed as *-er-∅* and *-er-s*. Finally, the algorithm prunes redundant or subsumed

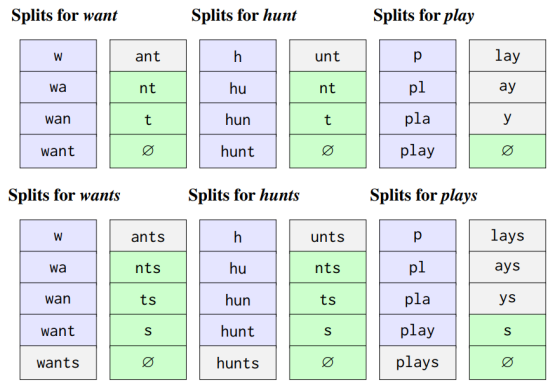

Figure 1: All possible binary splits for *want*, *hunt*, *play*, *wants*, *hunts* and *plays*. Boxes in purple indicate recurring roots, boxes in green stand for recurring suffixes. The paradigm that best accounts for the six lexical items is the one formed by productive roots *want*, *hunt* and *play*, and the productive suffixes -∅ and-*s*.

paradigms and ranks the remaining ones using a support score (adapted from Goldsmith, 2001): for any paradigm P with root set R and suffix set S, the support score is defined as

$$\text{Score}_P = \log_2(|R|) \times \log_2(|S|). \quad (4)$$

Words that do not belong to any paradigm are assigned to a "residual" paradigm, preventing spurious segmentations. During tokenization, ParFind first attempts to match a word against known paradigms and segment it accordingly; if no exact match is found, a fallback strategy matches the longest known suffix to recover partial structure.

All roots and suffixes from the paradigms, including those of the residual paradigm, are assigned unique token IDs. The vocabulary size obtained through this procedure was explicitly set to 30K.

4.3 Model Architecture and Training

We use GPT-2 as our base model to align with the BabyLM baselines and previous submissions. Unless stated otherwise, we train separate models for each tokenizer in both the strict and strict-small tracks.

Architecture. Our GPT-2 implementation follows the "base" configuration with 12 transformer layers ($n_{\text{layer}} = 12$), hidden size $n_{\text{embd}} = 768$, and 12 self-attention heads ($n_{\text{head}} = 12$). Each model has a context window of $n_{\text{positions}} = 1024$ tokens. This architecture yields approximately 110M trainable parameters. For fair comparison across tokenizers, we keep the non-embedding parameters fixed and adjust only the input embedding layer to accommodate the vocabulary size of each tokenizer.

Training procedure. Models are trained using the official BabyLM recipe. We adopt the following hyper-parameters:

- **Sequence length**: 512 tokens per example.
- **Batch size**: 16 examples.
- **Optimiser and learning rate schedule**: AdamW with a base learning rate of 5×10^{-5}, linear warm-up over the first 2,000 steps and weight decay of 0.01.
- **Training steps**: 200,000 steps (roughly 10 epochs over the strict-small data).
- **Gradient clipping**: max norm of 1.0.

4.4 Results

We evaluate both the tokenizers and the trained language models.

Morphological segmentation. For each tokenizer, we segment the SIGMORPHON benchmark words and compute precision, recall, F_1 and Levenshtein distance against the gold morpheme boundaries, following the SIGMORPHON 2022 evaluation procedure. These scores allow us to quantify how well each tokenizer captures linguistically meaningful units.

Results are shown in Table 2. The best performance on this benchmark is reached by ParadigmFinder, with an F1 score of 33.99, followed by MorPiece (F1 = 26.80), BPE (F1 = 23.50) and finally SylliTok (F1 = 14.98). This ordering is consistent with our expectations: both ParadigmFinder and MorPiece explicitly target morphemes as the fundamental units of segmentation, albeit in fully unsupervised ways, and therefore align more closely with the gold morphological boundaries. In contrast, BPE optimises for compression rather than linguistic structure, and SylliTok splits on syllables, a unit that often does not coincide with morpheme boundaries in English.

Tokenizer	Avg. Lev. Dist.	Prec	Rec	F1
BPE	2.08	21.03	26.62	23.50
MoP	1.96	24.54	29.52	26.80
SillyTok	2.77	12.45	18.81	14.98
ParFind	**1.24**	**38.99**	**30.12**	**33.99**

Table 1: Tokenizers' evaluation on SIGMORPHON using BabyLM 10M as training corpus.

BabyLM evaluation. We use the official BabyLM 2025 evaluation pipeline to assess our custom tokenizers when paired with a standard GPT-2 model architecture. These tasks collectively probe a wide range of linguistic and cognitive abilities—from syntactic acceptability and morphological generalisation to world knowledge, entity state tracking and alignment with human reading behaviour—providing a comprehensive evaluation of our tokenizers and models. We refer to the Appendix for a detailed description of the various tasks.

In Table 2, we report the macro-averaged score for each section of the benchmark. We compare the performance of our models with that of the challenge's baseline GPT-2 model with the BPE tokenizer. The results show no single tokenizer dominating across all tasks, but rather a complementary pattern that reflects the different linguistic biases each segmentation strategy encodes.

SylliTok performs surprisingly well on comprehension-oriented tasks: it achieves the best scores on BLiMP Supplement (58.8) and GLUE (58.1), while matching BPE performance on EWoK (≈49.9). However, it shows only a weak positive correlation with human judgements on WUG_ADJ (33.1) and a negative correlation on WUG_PAST (-29.4), highlighting the limits of a syllable-based representation when it comes to morphosyntactic generalisation.

MorPiece, which segments words into morphemes, offers a different trade-off: it improves semantic and discourse tasks—scoring higher on COMPS (55.8), EWoK (50.6) and by far the best on Entity Tracking (64.4) and WUG_PAST (12.1)—but it trails BPE on BLiMP and BLiMP Supplement and yields weaker results on WUG_ADJ (37.6) and AoA (-25.6), suggesting that morphological segmentation alone does not uniformly translate to improved performance.

ParadigmFinder achieves the best scores on the semantic task COMPS (56.6) and on BLiMP Supplement, matching SylliTok performance (58.8). It also yields the best AoA score (16.3), indicating a degree of alignment with developmental learning patterns of words. The surprisingly low performance on WUG_ADJ (-43.1) may be attributed to the difficulty of recognising multiple derivational suffixes.

BPE, while linguistically shallow, remains a strong baseline. It leads on BLiMP (66.4) and WUG_ADJ (66.1), showing that purely frequency-driven segmentation can sometimes outperform more linguistically motivated methods. Nonetheless, its generally moderate scores across the other tasks confirm that frequency alone is insufficient to consistently capture the kinds of regularities targeted by BabyLM with a small token budget (10M).

5 Discussion

The results indicate that introducing linguistically informed tokenizers does not lead to clear improvements on the more traditional grammar-oriented sections of the BabyLM benchmark. On BLiMP, for instance, all models perform at a similar level, with BPE in fact yielding the highest score. Likewise, on BLiMP Supplement, the differences are small, with ParadigmFinder and SylliTok only slightly surpassing BPE. This suggests that morphologically and syllable-aware tokenizations do not provide systematic advantages on syntactic acceptability judgments, at least under the strict 10M training budget.

More interestingly, gains appear in tasks that require richer semantic generalisation and discourse tracking. Both SylliTok and ParadigmFinder equate BPE on EWoK, while MorPiece slightly outperforms it. All of our models also surpass it on COMPS and Entity Tracking, pointing to improvements in comprehension-oriented evaluation. In particular, MorPiece achieves the highest score on Entity Tracking, highlighting the potential of morpheme-based segmentation for tasks that demand sensitivity to discourse-level dependencies. ParadigmFinder, on the other hand, shows competitive results on semantic composition (COMPS) and also exhibits a modest advantage in word Age of Acquisition (AoA), suggesting that a paradigm-based segmentation may capture aspects of lexical development more effectively than frequency-based subword units.

These results align with the findings from the intrinsic evaluation on the SIGMORPHON segmentation benchmark. There, ParadigmFinder and MorPiece achieved the best correspondence to morpheme boundaries, while BPE and SylliTok lagged behind. The parallel between segmentation accuracy and downstream comprehension/discourse gains suggests that morphological faithfulness in tokenization may indeed translate into advantages for meaning-sensitive tasks, even if not for purely grammatical ones.

Model	BLiMP	BLiMP Suppl.	COMPS	EWoK	Eye Track.	SPR	Entity Track.	WUG_ADJ	WUG_PAST	GLUE	AoA
GPT-2 + BPE	**66.4**	57.1	51.7	49.9	**8.7**	**4.3**	13.9	**66.1**	-5.0	55.9	11.7
GPT-2 + MoP	63.5	52.6	55.8	**50.6**	1.2	0.7	**64.4**	37.6	**12.1**	57.7	-25.6
GPT-2 + SylliTok	63.1	**58.8**	55.3	49.9	0.9	0.1	33.9	33.1	-29.4	**58.1**	-31.7
GPT-2 + ParFind	65.2	**58.8**	**56.6**	49.4	0.1	0.3	21.0	-43.1	-2.6	57.8	**16.3**

Table 2: Results of the BabyLM tasks evaluation of the baseline GPT-2 model trained using different tokenization strategies.

6 Conclusion

In this work, we evaluated several tokenizers designed to approximate linguistic units and tested them both in isolation (via the SIGMORPHON 2022 morpheme segmentation benchmark) and when paired with GPT-2 on the BabyLM 2025 evaluation suite. The findings indicate that while linguistically motivated tokenizers do not consistently outperform BPE on grammar-focused benchmarks, they offer complementary benefits on tasks targeting comprehension, discourse tracking, and developmental plausibility.

Taken together, the results reinforce our threefold perspective on tokenization: **(i) modeling**, since the segmentation choice directly affects the inductive biases available to the language model; **(ii) compression**, since different strategies vary in how efficiently they reduce entropy and distribute representational resources; and **(iii) morphology**, since tokenization determines the extent to which models can access and exploit the systematic structure of words. The SIGMORPHON results demonstrate that more morphology-aware tokenizers are indeed closer to humanlike segmentation, and the BabyLM evaluation reveals that this morphological consistency carries over into improvements in meaning-sensitive tasks.

Future work should expand evaluation to multiple languages, integrate hybrid tokenization strategies, and further investigate the alignment between human morphological acquisition and artificial segmentation methods. Ultimately, our findings suggest that tokenization should be treated not as a fixed preprocessing step, but as a substantive modeling decision with theoretical and practical consequences.

7 Limitations

Our experiments were conducted on the *strict-small* BabyLM corpus (10K tokens) rather than the full *strict* version (100K tokens). A direct comparison with models and tokenizers trained on the larger corpus would be essential to assess how data scale influences both tokenization quality and downstream performance.

Furthermore, we restricted our evaluation to a single baseline architecture (GPT-2). While this choice allowed for controlled comparisons across tokenization strategies, future work should test the generality of our findings across models of different sizes and architectures.

Finally, the relation between tokenization and compression remains to be explored in greater depth. In particular, future work should incorporate an explicit Minimum Description Length (MDL) metric (Goldsmith, 2001) to quantify how efficiently each tokenizer represents linguistic structure.

8 Ackowledgments

We acknowledge financial support under the National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.1, Call for tender No. 104 published on 2.2.2022 by the Italian Ministry of University and Research (MUR), funded by the European Union – NextGenerationEU – Project Title T-GRA2L: Testing GRAdeness and GRAmmaticality in Linguistics (202223PL4N) – CUP I53D23003900006 - Grant Assignment Decree No. 104 adopted on the 2nd February 2022 by the Italian Ministry of Ministry of University and Research (MUR). PI: CC. This work contains simulations carried out on the High Performance Computing DataCenter at IUSS, co-funded by Regione Lombardia through the funding programme established by Regional Decree No. 3776 of November 3, 2020

References

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 Pho-*

netics, Phonology, and Morphology, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Jean Berko. 1958. The child's learning of english morphology. *Word*, 14(2-3):150–177.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2025. Small language models also work with small vocabularies: Probing the linguistic abilities of grapheme- and phoneme-based baby llamas. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6039–6048, Abu Dhabi, UAE. Association for Computational Linguistics.

Tyler A. Chang and Benjamin K. Bergen. 2022. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16.

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.

Andrea Gregor de Varda, Marco Marelli, and Simona Amenta. 2024. Cloze probability, predictability ratings, and computational estimates for 205 English sentences, aligned with existing EEG and reading time data. *Behavior Research Methods*, 56(5):5190–5213.

Achille Fusco, Matilde Barbini, Maria Letizia Piccini Bianchessi, Veronica Bressan, Sofia Neri, Sarah Rossi, Tommaso Sgrizzi, and Cristiano Chesi. 2024. Recurrent networks are (linguistically) better? an (ongoing) experiment on small-LM training on child-directed speech in Italian. In *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*, pages 382–389, Pisa, Italy. CEUR Workshop Proceedings.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Omer Goldman, Avi Caciularu, Matan Eyal, Kris Cao, Idan Szpektor, and Reut Tsarfaty. 2024. Unpacking tokenization: Evaluating text compression and its correlation with model performance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2274–2286, Bangkok, Thailand. Association for Computational Linguistics.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

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.

Valentin Hofmann, Leonie Weissweiler, David R. Mortensen, Hinrich Schütze, and Janet B. Pierrehumbert. 2025. Derivational morphology reveals analogical generalization in large language models. *Proceedings of the National Academy of Sciences*, 122(19):e2423232122.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi U. Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian C. Paulun, Maria Ryskina, Ekin Akyürek, Ethan G. Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2025. Elements of world knowledge (EWoK): A cognition-inspired framework for evaluating basic world knowledge in language models. *Transactions of the Association for Computational Linguistics*, 13:1245–1270.

Haris Jabbar. 2023. Morphpiece: A linguistic tokenizer for large language models. *arXiv preprint arXiv:2307.07262*.

Najoung Kim and Sebastian Schuster. 2023. Entity tracking in language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3835–3855, Toronto, Canada. Association for Computational Linguistics.

Constantine Lignos and Charles Yang. 2016. *Morphology and Language Acquisition*, page 765–791. Cambridge Handbooks in Language and Linguistics. Cambridge University Press.

Jacques Mehler, Peter Jusczyk, Ghislaine Lambertz, Nilofar Halsted, Josiane Bertoncini, and Claudine Amiel-Tison. 1988. A precursor of language acquisition in young infants. *Cognition*, 29(2):143–178.

Kanishka Misra, Julia Rayz, and Allyson Ettinger. 2023. COMPS: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2928–2949, Dubrovnik, Croatia. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2025. The impact of token granularity on the predictive power of language model surprisal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4150–4162, Vienna, Austria. Association for Computational Linguistics.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. 2025. Byte latent transformer: Patches scale better than tokens. In *Proceedings*

of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9238–9258, Vienna, Austria. Association for Computational Linguistics.

Bharath Raj S, Garvit Suri, Vikrant Dewangan, and Raghav Sonavane. 2025. When every token counts: Optimal segmentation for low-resource language models. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 294–308, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Michelle Sandoval, Dianne Patterson, Huanping Dai, Christopher J. Vance, and Elena Plante. 2017. Neural correlates of morphology acquisition through a statistical learning paradigm. *Frontiers in Psychology*, Volume 8 - 2017.

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.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. SuperGLUE: a stickier benchmark for general-purpose language understanding systems. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. 2023a. Call for papers – the BabyLM challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2301.11796.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023b. 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.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Valentin Hofmann, Anjali Kantharuban, Anna Cai, Ritam Dutt, Amey Hengle, Anubha Kabra, Atharva Kulkarni, Abhishek Vijayakumar, Haofei Yu, Hinrich Schuetze, Kemal Oflazer, and David Mortensen. 2023. Counting the bugs in ChatGPT's wugs: A multilingual investigation into the morphological capabilities of a large language model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6508–6524, Singapore. Association for Computational Linguistics.

Hongzhi Xu, Mitchell Marcus, Charles Yang, and Lyle Ungar. 2018. Unsupervised morphology learning with statistical paradigms. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 44–54, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Charles Yang. 2016. *The price of linguistic productivity: How children learn to break the rules of language*. MIT press.

A Appendix

The BabyLM 2025 benchmark is structured as follows:

Zero-shot linguistic preference tasks.

- **BLiMP** — The Benchmark of Linguistic Minimal Pairs (Warstadt et al., 2020) is a challenge set designed to probe what language models know about core grammatical phenomena in English. It comprises 67 automatically generated sub-datasets, each containing 1,000 minimal sentence pairs that isolate a particular syntactic, morphological or semantic contrast. Models are scored by whether they assign higher probability to the grammatical sentence in each pair.

- **BLiMP Supplement** — An extension of BLiMP tailored for BabyLM (Warstadt et al., 2023b). The supplement introduces additional contrasts (e.g. lexical and morphological judgments) not covered in the original BLiMP. As with BLiMP, models must prefer the acceptable sentence in each minimal pair.

- **EWoK** — The Elements of World Knowledge framework (Ivanova et al., 2025) evaluates basic world-modeling abilities by asking models to judge which of two context/target pairs is more plausible. The EWoK-CORE-1.0 dataset contains 4,374 items spanning 11

knowledge domains (from social interactions to spatial relations). Minimal pairs are constructed so that only one sentence aligns with commonsense world knowledge.

- **Entity Tracking** — Based on the task of Kim and Schuster (2023), this evaluation tests a model's ability to keep track of entities and their states as a text unfolds. A model is given an initial description of an entity and a series of state-changing operations and must assign higher probability to the correct continuation that reflects the entity's final state. In the BabyLM pipeline this task is evaluated in a zero-shot setting by computing sentence logit scores.

- **Derivational Morphology (WUG_ADJ)** — Following Hofmann et al. (2025), this task tests morphological generalisation via an adjective-nominalisation "wug" experiment. Models see nonce adjectives (e.g. *daxen*) and must decide whether the corresponding noun uses the suffix *-ity* or *-ness*. Performance is measured by the correlation between model probabilities and human judgements.

- **WUG_PAST** — From Weissweiler et al. (2023), this hidden task evaluates how models generalise past-tense formation to nonce verbs. Models are presented with a novel verb and several possible past-tense forms; their probability distribution is correlated with human responses.

- **COMPS** — The Conceptual Minimal Pair Sentences dataset (Misra et al., 2023) tests whether language models know that properties of superordinate concepts are inherited by subordinate concepts. Sentences feature nonce words standing in hierarchical relations (e.g. a *lorp* is a type of *bim*); models must assign higher probability to the sentence that correctly inherits the property.

- **Cloze probability and reading time (Self-paced Reading and Eye-tracking)** — Adapted from de Varda et al. (2024), this benchmark links language model predictions to human reading times. The evaluation computes the increase in explained variance (R^2) in human eye-tracking measures with no spill-over effect and in self-paced reading with a one-word spillover. It assesses the alignment between model surprisal and human processing difficulty.

- **Age of Acquisition (AoA)** — Based on Chang and Bergen's (2022) methodology, this benchmark tracks word surprisal across training checkpoints to estimate when a model "acquires" each word. The resulting learning curves are fitted with sigmoid functions and correlated with human Age-of-Acquisition norms from the MacArthur–Bates Communicative Development Inventory.

Fine-tuning tasks.

- **(Super)GLUE** — The General Language Understanding Evaluation benchmark (Wang et al., 2018, 2019) comprises a suite of natural-language understanding tasks (e.g. sentiment analysis, paraphrase detection, natural-language inference). In BabyLM it is used to assess models' ability to generalise via supervised fine-tuning on tasks such as MNLI, SST-2, QQP and QNLI.