Contextual morphologically-guided tokenization for pretrained Latin BERT models

Anonymous Authors[1]

Abstract

Tokenization is a critical component of language model pretraining, yet standard tokenization methods often prioritize information-theoretical goals like high compression and low fertility rather than linguistic goals like morphological alignment. In fact, they have been shown to be suboptimal for morphologically rich languages, where tokenization quality directly impacts downstream performance. In this work, we investigate morphologically-aware tokenization for Latin, a morphologically rich, medium-resource language. For both the standard WordPiece and Unigram Language Model (ULM) tokenization models, we propose two variations: one seeded with known morphological suffixes in the tokenizer vocabulary, and another using contextual pretokenization with a language-specific, lexicon-based morphological analyzer. From each learned tokenizer, we pretrain Latin BERT and evaluate its performance on POS and morphological feature classification. We find that morphologically-guided tokenization improves overall performance (e.g., 36% relative error reduction for morphological feature accuracy), with particularly large gains for specific, morphologically-signalled features (e.g., 54% relative error reduction for tense prediction). Our results highlight the utility of morphological linguistic resources to improve language modeling for morphologically complex languages.

1. Introduction

Tokenization is the first step in Large Language Model (LLM) pretraining pipeline, making it the foundation upon which model performance rests. A common assumption is that tokenizers should maximize compression and minimize fertility (Schmidt et al., 2024). However, recent research has challenged this view, particularly in the context of morphologically rich and lower-resource languages. Studies have shown that existing tokenization methods exhibit low morphological alignment (Hsu et al., 2023; Bostrom & Durrett, 2020; Erkaya, 2022; Libovick'y & Helcl, 2024), which can negatively impact downstream performance. In this work, we investigate tokenization strategies for Latin, a morphologically rich, medium-resource language[1] with a long scholarly tradition, and moreover one with specific interest in word endings and other aspects of word formation, making it an informative test case.

We hypothesize that incorporating morphological knowledge into tokenization will improve both morphological alignment and downstream performance. While prior work has explored morphologically-aware tokenizers, they often focus on high-resource and/or morphologically simple languages (Hofmann et al., 2021; Hsu et al., 2023; Bostrom & Durrett, 2020) or employ acontextual, unsupervised morphological analyzers such as Morfessor (Creutz & Lagus, 2005). Furthermore, evaluations in this area have not examined fine-grained morphological feature prediction, which should better capture whether a morphologically-aligned tokenization helps the language model's contextual embeddings capture its linguistic content.[2]

Beyond its computational implications, this question is of particular interest within Latin linguistic and philological research. The role of word endings in meaning construction has been central to Latin scholarship for centuries, making it crucial to empirically test whether morphology-informed tokenization aligns with these long-standing linguistic intuitions.

To test our hypothesis, we experiment with three types of tokenizers:

1. **Baseline:** Standard WordPiece and Unigram Language Model (ULM)
2. **Low-guidance approach:** Seeding morphological suf-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]Between 100M and 1B tokens (Chang et al., 2024)

[2]Much prior work attains a partial view of this by evaluating on POS tagging, a significantly coarser version of the problem.

fixes into the tokenizer vocabulary. This approach is lightweight, only requiring a predefined list of suffixes.

3. **High-guidance approach:** Pre-Tokenization using a morphological analyzer. Unlike prior work, we disambiguate the analyses based on POS information, making our method context-aware.

One of the unique advantages of working with Latin is the availability of expert-curated morphological and lexical resources—benefiting from long-standing philological knowledge about the role of word endings in constructing meaning. In this work, we leverage Lemlat (Passarotti et al., 2017), a Latin-specific lemmatizer and morphological analyzer, to implement both our low and high-guidance tokenization methods.

For evaluation, we pretrain Latin BERT models (Devlin et al., 2019) using each tokenizer and finetune them for POS tagging and morphological feature classification. To our knowledge, no prior studies have evaluated their morphologically aware tokenization methods using morphological feature classification as a downstream task. We observe larger performance gains for this task than POS tagging, likely due to the granularity of the labels.

2. Related Work

2.1. Tokenization

Tokenization plays a crucial role in language model pretraining, yet its impact on morphologically rich languages remains an active area of investigation. Studies have increasingly questioned whether widely used tokenization methods such as Byte Pair Encoding (BPE) (Sennrich et al., 2016), Unigram Language Model (ULM) (Kudo, 2018), and WordPiece (Schuster & Nakajima, 2012) sufficiently capture morphological structure, and whether this matters for downstream performance.

Several studies suggest that aligning English BPE and ULM tokenizers' segmentations with gold-standard morphological boundaries can enhance downstream performance on sentiment and topic classification (Hofmann et al., 2021), zero-shot summarization and retrieval (Hsu et al., 2023), QA, MNLI, and NER (Bostrom & Durrett, 2020).

Beyond English, studies on morphologically rich languages provide further evidence for the benefits of morphology-aware tokenization. Rule-based approaches have shown improvements in Romanian NLP tasks (Vasiu & Potolea, 2020), and pre-tokenization methods using 1) a language-specific morphological analyzer in Turkish (Erkaya, 2022), Kinyarwanda (Nzeyimana & Niyongabo Rubungo, 2022), or 2) with Morfessor in eight languages (Libovick'y & Helcl, 2024) have all yielded downstream performance gains. Post-training strategies have also been effective; for instance, modifying existing BPE vocabularies improved token-level tasks in English, Dutch, and German (Bauwens & Delobelle, 2024).

While most morphologically-aware tokenization methods rely on static rules or unsupervised segmentation, some studies have experimented with adding contextual information. Yehezkel & Pinter (2023) introduced SaGe, a tokenizer whose vocabulary construction method closely resembles ULM but incorporates a SkipGram objective to refine vocabulary selection. This approach improved performance on English GLUE and NER, and Turkish Inference and NER.

Despite strong evidence supporting the benefits of morphological alignment in tokenization, some studies have produced mixed or contradictory results. Schmidt et al. (2024) disprove the assumption that compression is the primary determinant of tokenizer quality, in line with our own intuition, but found that the morphologically informed tokenizer they tested, SaGe, was not always the top performer.

Other studies directly oppose the hypothesis that morphological alignment is beneficial. Toraman et al. (2022) found no improvements in Turkish NLP tasks when pre-tokenizing with a morphological analyzer, though they noted that errors in the analyzer itself may have influenced results. More broadly, Arnett & Bergen (2025) argued that morphological alignment is not a key factor in tokenization quality, emphasizing instead that dataset size and quality are more important.

The existing literature provides strong, though not unanimous, evidence that morphologically-aware tokenization can improve NLP performance, particularly for morphologically rich languages. However, prior studies have largely focused on a limited set of downstream tasks—such as POS tagging and NER—which may not fully expose the benefits of morphologically-aligned tokenization. Our work extends this research by evaluating multiple tokenization strategies for Latin, including both light and high-guidance approaches. Additionally, we introduce morphological feature classification as an alternative downstream evaluation metric, hypothesizing that the fine-grained morphological feature values will reveal improvements that are less evident in coarse-grained POS tagging.

2.2. Morpheme Segmentation

Morpheme segmentation has been widely studied as an independent NLP task, distinct from its potential applications in tokenization and language model pretraining. One of the most prominent efforts in this area is the SIGMORPHON shared task on morpheme segmentation (Batsuren et al., 2022), which provided segmentation data for nine languages, including Latin. This task included both acon-

textual and contextual segmentation challenges; however, Latin was excluded from the contextual segmentation track. While the availability of segmentation resources for these nine languages is valuable, the dataset is too small to support pretraining efforts. Moreover, many morphological datasets, including those used in SIGMORPHON, are constructed through automatic extraction from sources such as Wiktionary, introducing data quality concerns. For example, Gorman et al. (2019) highlight extensive extraction errors in the dataset for the 2017 CoNLL-SIGMORPHON shared task for Morphological Reinflection (Cotterell et al., 2017).

Latin is unique within morpheme segmentation research due to its rich morphological tradition and availability of high-quality, expert-curated resources. Unlike many other languages, Latin benefits from centuries of linguistic study focused on word formation and morphological structure (Diederich, 1936; Pellegrini et al., 2021).

Our work builds on the precisely curated morphological resources available for Latin, incorporating linguistic knowledge at the morpheme level in a way that is not feasible for many other languages. We argue that context-aware morphological tokenization–though requiring significant language-specific effort–has the potential to bridge the gap between linguistic theory and modern NLP.

2.3. Language Modeling

Several studies have demonstrated the feasibility of pretraining transformer-based models for low-resource languages. Ogueji et al. (2021) introduced AfriBERTa, a multilingual BERT model trained on various low-resource African languages using a corpus of approximately 1GB, comparable in size to ours. Their model outperforms massively multilingual models like mBERT and XLM-R (Conneau et al., 2020) on NER and text classification. Similarly, Chang et al. (2024) systematically evaluated the relationship between model size (from tiny to small) and data size across both monolingual and multilingual GPT-2 models (Radford et al., 2019).

Prior work in Latin language modeling has produced several pretrained, transformer-based Latin models. LaBERTa (Riemenschneider & Frank, 2023) was trained on 165M words from Corpus Corporum (Roelli, 2014),[3] a kind of super-repository of available smaller digitized Latin text repositories. Another Latin RoBERTa model (Ströbel, 2022; Liu et al., 2019) was trained on a 390M token corpus also derived from Corpus Corporum. Finally, LatinBERT (Bamman & Burns, 2020) was trained on a larger corpus (642.7M words), though a significant portion originated from noisy OCR-processed Latin texts from the Internet Archive. Its cleaner subset contained 81.6M words.

[3] https://mlat.uzh.ch/

Our work differs from these prior efforts in Latin language modeling in three ways. First, we train small models (29M parameters) rather than "base"-sized architectures (110M parameters), aligning with broader research on low-resource pretraining. Second, our training corpus, totaling 195M words (1GB), is larger than the clean subset used in LatinBERT though smaller than the dataset used for Latin RoBERTa. Finally, we experiment with various morphologically-aware tokenization strategies, whereas existing Latin language models use baseline WordPiece and BPE.

3. Background: Tokenization

Schmidt et al. (2024) conceptualize tokenization as a three-step process: 1) pretokenization, which applies an initial set of rules to define processing units–typically by segmenting on whitespace; 2) vocabulary construction or training, where subword units are learned; 3) segmentation or decoding, which determines how input text is tokenized based on the trained vocabulary. This framework helps highlight the different places where morphological guidance can be introduced, instead of viewing tokenizers as indivisible systems. We experiment with modifications to two widely-used Huggingface tokenizer implementations.

3.1. WordPiece Tokenization

BPE and WordPiece are tokenizers with similar training algorithms. While BPE is widely used, prior work finds it suboptimal (Bostrom & Durrett, 2020; Erkaya, 2022), so our experiments focus on WordPiece. For clarity, we overview both in this section.

Pretokenization for both tokenizers is typically done by splitting on whitespace and punctuation. Given a list of (pretokenized) strings and a desired final vocabulary size, an initial subword vocabulary is constructed from all unique characters. Subword types are iteratively merged until the desired vocabulary size is reached. For WordPiece, the subword bigram with the highest pointwise mutual information (PMI) (Bouma, 2009) is chosen. Its two subwords are merged into a new, single subword and added to the vocabulary. For BPE, the bigram with the highest frequency is chosen rather than highest PMI.

To tokenize new text, WordPiece performs greedy left-to-right decoding, whereas BPE applies merge rules in the order learned during training.

3.2. Unigram Language Model Tokenization

The Unigram Language Model (Kudo & Richardson, 2018) infers the most likely segmentation for a word, using the Viterbi algorithm. For learning, after initial pretokeniza-

tion,[4] the model starts with a large vocabulary of all substrings in the corpus. Subwords are iteratively pruned in order to maximize the unigram likelihood of the corpus until the desired vocabulary size is reached.

4. Method

4.1. Data

Tokenizer and Pretraining Data We train our tokenizers and pretrain our BERT models on the same data used to train the static floret vectors (Boyd & Warmerdam, 2022) used in LatinCy, a spaCy pipeline for Latin (Burns, 2023; Honnibal & Montani, 2017). It is 1.08GB, containing 13.5M sentences and 195M whitespace-separated words.[5] This is comparable to the pretraining data size of other Latin transformer models; for example, Riemenschneider & Frank (2023) trained LaBERTa on 167.5M words.

4.2. Morphologically-Enhanced Tokenizers

We add morphological guidance to both ULM and WordPiece tokenizer models, implemented by modifying HuggingFace's implementations of each (Wolf et al., 2020). We create and evaluate three tokenizer variations: MorphSeeding, MorphPreTokenization (acontextual), and MorphPreTokenization (contextual).

MorphSeeding We create a list of 480 morphological suffixes sourced from Lemlat, a type-level lemmatizer and morphological analyzer for Latin (Passarotti et al., 2017). For our purposes, we define morphological suffixes as all segments of a word which are not the first (root/stem) segment.

Then, we modify the ULM and WordPiece trainers to bias them to prefer segmenting with this list of suffixes. For WordPiece, all suffixes are added to the initial vocabulary with the continuing subword prefix "##" prepended. Since WordPiece's vocabulary construction is bottom-up, once added to the vocabulary a subword cannot be removed. For ULM, all suffixes are added to the initial vocabulary, and for decoding, their log-probabilities are upweighted by a fixed amount[6] in the lattice. Suffixes are not allowed to be removed from the vocabulary.

MorphPreTokenization We analyze all unique words in our corpus with Lemlat. For each word, Lemlat returns a list of possible analyses that include the word's segmentation into morphemes, as well as its lemma, declension or conjugation, part of speech, morphological features, and derivational affixes. We only utilize the segmentation and POS.

We then presegment these morphemes in our corpus, so that during tokenizer training, it will never merge Lemlat-provided morphemes. We experiment with both contextual and acontextual segmenters.

For acontextual pretokenization, we simply use the segmentation in the first analysis given by Lemlat. This follows the type-level focus of previous work, either with language-specific morphological analyzers (Toraman et al., 2022; Erkaya, 2022; Nzeyimana & Niyongabo Rubungo, 2022) or the unsupervised Morfessor model (Creutz & Lagus, 2005; Libovick'y & Helcl, 2024).

But in many instances, there exists ambiguity over a word's segmentation, which can be resolved with contextual information about its grammatical role. Thus, we construct a contextual morphological segmenter by first running an off-the-shelf part-of-speech tagger on the corpus, and filtering Lemlat's output to an analysis with a matching POS tag.[7]

We tag our corpus with LatinCy (Burns, 2023). It uses the Latin UD Treebanks' tagset, which differs from Lemlat's. We create a mapping between the tag systems (Table 4), and a protocol for selecting a word's segmentation when the POS tags do not match:

- If Lemlat only gives one unique possible segmentation, use that one (occurs in 1.2% of words in UD treebanks).
- If Lemlat gives multiple possible segmentations but none match the predicted POS, do not segment the word (occurs in 0.028% of words in UD treebanks).

A word's tag usually disambiguates the segmentation, but in rare cases, one word may have multiple analyses with the same POS tag, due to multiple possible lemmas or morphological features. In these cases, we choose one segmentation based on the following criteria:

- If the candidate segmentations have the same number of subwords, choose the one with the longer suffix (i.e. out of the adjective [*adversar*, *-i*] versus infinitive verb [*advers*, *-ari*] choose the latter).
- If the candidate segmentations have a different number of subwords, choose the one with more subwords (i.e. out of participle [*inordin*, *-at*, *-o*] and imperative [*inordin*, *-ato*], choose the former).

[4] Pretokenization is also done by splitting on whitespace. In the HuggingFace implementation, punctuation is not split on.

[5] The ULM tokenizers are trained on 5% of this corpus. See §9.

[6] During initial experiments, we found a weight of 0.5 to strike a good balance of encouraging these suffixes to be chosen more often when decoding, while not increasing fertility to an unreasonable degree.

[7] Interestingly, for some tasks the approach may seem circular: a predicted POS tag helps guide the LLM tokenization, and thus the eventual LLM contextual representation used to predict, for example, a POS tag. Investigating how initial tagging errors propagate would be interesting future work.

This type of conflict occurred in 4.55% of all Lemlat analyses of unique words in the Latin treebanks.[8]

This results in four morphoglical pretokenization-based tokenizers—for each model class (ULM and WordPiece), there is both acontextual and contextual presegmentation.

We implement changes to the tokenizers to accommodate presegmentation. For ULM, the only modification is to add a new pre-tokenization step which splits on our morpheme symbol, allowing morphological suffixes to be treated as continuing subwords. Due to how the ULM vocabulary is constructed, it is possible that the suffixes will be split into multiple subwords, just like the root.

WordPiece requires modification to its trainer, not just the pretokenizer; for implementation details, see §A.2. Unlike ULM, once the suffix subword is added to the WordPiece vocabulary, it will remain unchanged, neither split or merged.

4.3. Tokenizer Evaluation

Several metrics have been proposed to assess tokenizer quality.

Renyi entropy (Zouhar et al., 2023) measures the uniformity of token frequency distributions. However, Schmidt et al. (2024) found that it correlates with Corpus Token Count, which they argue is a poor predictor of downstream performance.

A more linguistically motivated approach is morphological alignment with a gold reference segmentation, which assumes that having "meaningful" subword units improves downstream task performance. Various metrics have been introduced to quantify this.

MorphScore (Arnett & Bergen, 2025) assigns a score of 1 if a tokenizer correctly segments at a specific morpheme boundary, regardless of other boundaries in the word, and 0 otherwise. Unlike other measures, it excludes words that remain unsegmented.

Suffix precision, recall, and f1 (Erkaya, 2022) evaluate how well a tokenizer captures *suffix* boundaries specifically. Subword boundary precision, recall, f1 (Bostrom & Durrett, 2020) which we adopt in this work, assess overall segmentation accuracy. In addition, we track exact matches between predicted and gold segmentations.

The reliability of these metrics depends on the quality of the gold standard segmentations. Many studies experiment with multiple languages and rely on morphological data scraped from Wiktionary. Gorman et al. (2019) highlight extensive errors in SIGMORPHON's morpheme reinflection data (Cotterell et al., 2017), demonstrating that such resources may introduce inconsistencies. This underscores the importance of carefully curating high-quality gold standards, which tends to be easier when focusing on a single language. When Latin scholar coauthors reviewed the SIGMORPHON segmentation dictionary alongside two open-source Latin morphological dictionaries (Lemlat and Whitaker's Words[9]), we judged Lemlat to be highest quality. Lemlat has also been shown to have better coverage of Latin word types and tokens than Words, and equivalent coverage to LatMor, a finite state transducer for Latin (Springmann et al., 2016).

To construct an evaluation set, we extract all unique (word, POS) pairs from the five Latin UD test sets, and segment them using Lemlat. In the acontextual setting, we ignore POS and consider the first segmentation given by Lemlat as the gold segmentation. In the contextual setting, we disambiguate Lemlat's analyses using the gold UD POS, choosing the gold segmentation as described in §4.2.

4.4. BERT Pretraining

We use Megatron-LM (Shoeybi et al., 2020) to pretrain eight small (29M parameters) BERT models (Devlin et al., 2019), following prior work in pretraining for low and medium-resource languages (Ogueji et al., 2021; Chang et al., 2024). See §A.3 for details on hyperparameters.

4.5. Downstream Tasks

Modeling To evaluate our pretrained models, we finetune them for POS tagging and morphological feature classification. We use a separate classification head for each morphological feature, the same architecture as Riemenschneider & Frank (2023)'s finetuned Greek model. See §A.4 for details on hyperparameters. **[stats on the tagsets? number of features? –MH] [Not a bad idea—but perhaps wait to see if a reviewer requests this specifically? –PJB]**

Metrics We report whole-string morphological accuracy, following the convention of Gamba & Zeman (2023) and Sprugnoli et al. (2022). This metric considers the model's prediction correct when every morphological feature is correctly predicted, indicating whether the model understands how all the morphological features fit together.

5. Results

5.1. Tokenizer Evaluation

We evaluate the morphological alignment of each tokenizer by comparing predicted segmentations to a gold-standard segmentations from Lemlat. As seen in Table 1, baseline WordPiece already exhibits relatively strong alignment with gold segmentations, outperforming ULM in this regard

[8] In both the UD treebanks and in LASLA (Denooz, 2004), a non-UD Latin treebank.

[9] https://latin-words.com/

Tokenizer Type	Model Class	Acontextual Exact Match	Contextual Exact Match
Baseline	ULM	.1387	.1076
MorphSeed	ULM	.1524	.1212
MorphPreTok (Actx.)	ULM	.7001	.6505
MorphPreTok (Ctx.)	ULM	.6544	.7188
Baseline	WP	.2267	.2012
MorphSeed	WP	.2273	.2018
MorphPreTok (Actx.)	WP	.8281	.7550
MorphPreTok (Ctx.)	WP	.7538	.8432

Table 1. Tokenizers' morphological segmentation accuracy, evaluated both on acontextual and contextual versions of segmentations

(+9.4% exact match against the gold contextual segementations).[10]

Introducing morphological pretokenization (MorphPreTok) significantly enhances alignment for both ULM and WordPiece, with exact matches exceeding 65% for all variants. This suggests that explicitly incorporating morphological information during pretokenization leads to segmentations that closely mirror linguistic ground truth.

By contrast, morphological suffix seeding (MorphSeed) provides only a modest improvement for ULM (+1.4% exact match against both acontextual and contextual gold segmentations), while having no effect on WordPiece's alignment. This suggests that while suffix seeding can nudge segmentation toward morphological boundaries, they are less effective than full pretokenization in enforcing linguistically coherent segmentations.

5.2. Downstream Performance

POS Across all models, morphological pre-tokenization (MorphPreTok) methods led to improvements in POS tagging accuracy, with gains ranging from 1.0-1.3% over the baseline models (Table 2). In contrast, morphological suffix seeding (MorphSeed) had little to no effect on POS accuracy.

Morphological Features For ULM models, contextual pre-tokenization resulted in a substantial overall morphological accuracy improvement, from 90% to 94%. Both acontextual and contextual pre-tokenization methods led to consistent performance gains across all morphological features, with Tense showing the largest improvement in macro F1 over the baseline (+14). MorphSeed, while beneficial for Degree and Tense, did not improve other features and in some cases led to slight regressions, ultimately yielding similar overall morphological accuracy to the baseline ULM model.

WordPiece models produced more mixed results, with no single method outperforming the baseline across all morphological features. The most substantial improvements were observed for Tense (all tokenizer variants) and Degree (for MorphPreTok variants; MorphSeed caused a 9.7 drop in Degree macro F1). Acontextual pre-tokenization led to slight gains in Mood, while MorphSeed improved Voice. However, overall morphological accuracy remained largely unchanged across all WordPiece models.

When evaluated on whole-string morphological accuracy, the ULM model with contextual pre-tokenization outperformed both the baseline WordPiece model and the best-performing WordPiece variation (MorphSeed). These findings suggest that high morphological guidance, particularly through contextual pre-tokenization, enhances model performance more effectively than morphological suffix seeding alone, with the greatest benefits observed in ULM-based models.

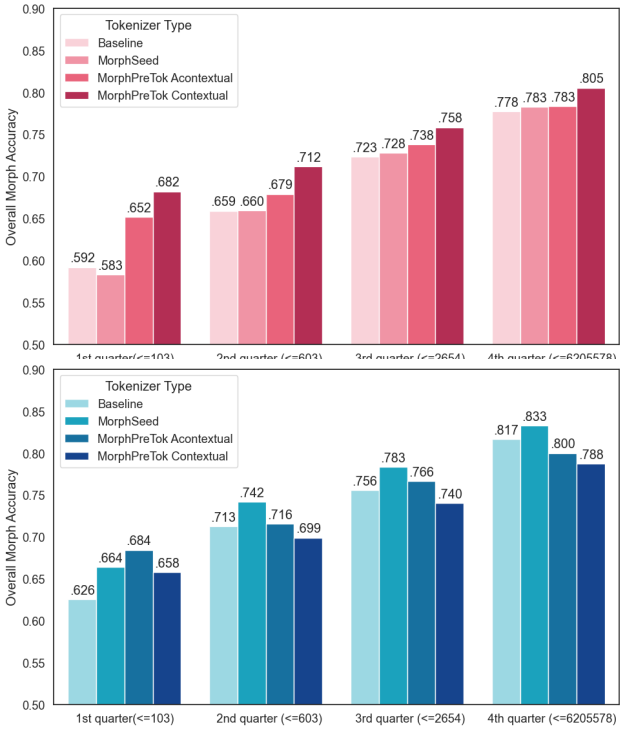

Figure 1. Word frequency in the pretraining corpus versus whole-string morphological accuracy, for ULM (top) and WordPiece (bottom).

Effect of word frequency Figure 1 shows the relationship between words' downstream morphological feature accuracy and their frequency in the pretraining corpus.[11]

For the rarest words ($\leq$ 103 occurrences in the pretraining corpus), nearly all morphology-guided tokenization methods improve overall morphological accuracy. The only ex-

[10] For complete evaluation metrics, see Table 3.

[11] Since this analysis excludes punctuation, numbers, and single-character words, overall accuracy appears lower than what is reported in Table 2.

Tokenizer Type	Model Class	POS Acc	Morph Acc	Per-Feature Macro F1							
				Case	Degree	Gender	Mood	Number	Person	Tense	Voice
Baseline	ULM	.9466	.9065	.6286	.8361	.8385	.6087	.9147	.9071	.7382	.8723
MorphSeed	ULM	+.0021	-.0041	+.0003	+.0193	+.0001	-.0024	-.0114	-.0085	+.0349	-.0051
MorphPreTok Acontextual	ULM	+.0115	+.0211	+.0109	+.0382	+.0439	+.0502	+.0161	+.0391	+.0678	+.0408
MorphPreTok Contextual	ULM	+.0132	+.0341	+.0423	+.0537	+.0591	+.0821	+.0374	+.0389	+.1476	+.0653
Baseline	WP	.9487	.9306	.6576	.9523	.8646	.6464	.9374	.9502	.7258	.8938
MorphSeed	WP	+.0013	+.0057	-.0008	-.0969	+.0120	+.0055	-.0029	-.0107	+.0852	+.0209
MorphPreTok Acontextual	WP	+.0103	+.0007	-.0038	+.0332	+.0062	+.0373	-.0139	-.0255	+.1037	+.0100
MorphPreTok Contextual	WP	+.0136	-.0080	-.0098	+.0076	-.0172	+.0064	-.0188	-.0278	+.1150	-.0073

Table 2. Downstream POS accuracy, whole-string morphological feature accuracy, and per-feature macro F1 scores. Performance is shown for the baseline of each model class, and for their morphologically-aware variants, the difference from that class's baseline.

ception is ULM MorphSeed, which experiences a 0.9% drop. Aside from this, all ULM-based models outperform the baseline across all frequency ranges. However, the performance gap narrows as word frequency increases, from a +9.0% improvement over the baseline for rare words to +2.8% for the most frequent words.

WordPiece-based models show a more variable trend. While all variants improve accuracy for the rarest words, some begin to degrade performance at higher frequencies. Contextual MorphPreTok sees a decline for words occurring $>$103 times, while Acontextual MorphPreTok degrades performance for words occurring $>$2,654 times. In contrast, MorphSeed consistently improves performance across all frequency ranges. The gap between it and the baseline shrinks from 3.9% (rarest words) to 1.6% (most frequent words), a trend similar to ULM.

6. Qualitative Analysis

WordPiece Improvements The word *scientur* (3rd person, plural, passive, future tense) demonstrates how WordPiece's left-to-right greedy decoding can lead to suboptimal segmentations that affect downstream predictions. The baseline and MorphSeed tokenizers maximize the length of the first subword, producing [*scient*, *-ur*]. Consequently, both models mispredict *scientur* as present tense, and the baseline model mispredicts singular, likely because *-ur* appears in singular, present-tense passive verbs (*-tur* being the canonical suffix). In contrast, when constrained to morpheme boundaries, both MorphPreTok models correctly segment *scientur* as [*sc*, *-ientur*], aligning with the gold Lemlat segmentation and enabling the correct predictions of plural and future tense.

ULM Improvements The opposite issue arises with the verb *impropero*. Its *-ero* ending resembles a future-tense marker, but it is actually a present-tense verb with the root *improper-*. Unlike WordPiece, which greedily selects *improper-* as the first subword, ULM's Viterbi decoding optimizes for globally probable segmentations and instead produces [*imp*, *-rop*, *-ero*]. This segmentation leads to a misclassification as future tense. By enforcing morpheme boundaries, the MorphPreTok variants segment *impropero* correctly as [*improper*, *-o*], aligning with the gold standard and yielding the correct present-tense prediction.

Regressions Morphologically guided tokenizers can sometimes over-rely on word endings at the expense of sentence-level context. For example, the adjective *intelligibilis* was segmented by the acontextual MorphPreTok WordPiece tokenizer as [*intelligibil*, *-is*], leading the model to predict its case as genitive due to *-is* being the genitive marker for third-declension nouns. However, in this instance, *intelligibilis* is nominative, as it modifies the nominative noun *species*. The baseline and MorphSeed WordPiece tokenizers, which left *intelligibilis* unsegmented, correctly predicted its case. This suggests that while morphological segmentation can improve alignment, it may also bias models toward surface-level suffix patterns, sometimes at the cost of contextual understanding.

7. Ongoing Work: Additional Downstream Tasks

We recognize that the improvements observed in POS and morphological feature tagging may not generalize to other downstream tasks, especially more semantic or sentence-level tasks. We are in the process of adding three new tasks: named entity recognition (NER) (Erdmann et al., 2016; 2019; Beersmans et al., 2023), word sense disambiguation (WSD) (Ghinassi et al., 2024; Lendvai & Wick, 2022), and authorship verification (Gorovaia et al., 2024).

8. Conclusion

We demonstrate that morphologically-guided tokenization improves downstream performance in Latin BERT models, particularly for features that are strongly tied to morphological structure. Across both WordPiece and ULM tokenization frameworks, incorporating morphological suffix seeding or contextual pre-tokenization enhances morphological feature classification, with significant gains in tense prediction and overall accuracy. These findings reinforce the importance

of linguistically-informed tokenization, especially for morphologically rich languages.

More broadly, our results highlight the need for continued investment in developing high-quality linguistic resources, particularly for lower-resource and morphologically complex languages, where data availability remains a key bottleneck.

9. Limitations

Our ULM tokenizers are trained on 5% of our pretraining corpus, whereas the WordPiece tokenizers are trained on the full dataset. When ULM tokenizers were trained on the full corpus, we observed pathological behavior, including high fertility and segmentations with many single-character subwords and low morphological alignment. Training on smaller datasets, and this 5% sample, yielded much more regular results, for reasons unclear to us; future work could examine if it is an implementation issue. We decided to train ULM tokenizers on a subset of the corpus, in order to have higher-quality tokenization and a fairer comparison to the WordPiece tokenizers. All BERT models were pretrained on the full corpus.

We pretrain small (29M parameter) BERT models. The performance gains we observed may not scale to larger models or other architecture types.

We only pretrain and finetune a single model per tokenizer, in order to reduce computational time and cost.

All our downstream tasks are at the token level, rather than the sentence or chunk level. Although other research has shown performance gains on sentence-level tasks from morphologically-aware tokenization, it may not improve results for Latin specifically.

Impact Statement

In this paper, we seek to advance the understanding of tokenization and language modeling for Latin, a morphologically rich, medium resource language. The methods discussed are potentially relevant to improving performance for other low-resource and under-studied languages.

References

Arnett, C. and Bergen, B. Why do language models perform worse for morphologically complex languages? In Rambow, O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B. D., and Schockaert, S. (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 6607–6623, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.441/.

Bamman, D. and Burns, P. J. Latin bert: A contextual language model for classical philology, 2020. URL https://arxiv.org/abs/2009.10053.

Batsuren, K., Bella, G., Arora, A., Martinovic, V., Gorman, K., Žabokrtský, Z., Ganbold, A., Dohnalová, Š., Ševčíková, M., Pelegrinová, K., Giunchiglia, F., Cotterell, R., and Vylomova, E. The SIGMORPHON 2022 shared task on morpheme segmentation. In Nicolai, G. and Chodroff, E. (eds.), *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pp. 103–116, Seattle, Washington, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.sigmorphon-1.11. URL https://aclanthology.org/2022.sigmorphon-1.11/.

Bauwens, T. and Delobelle, P. BPE-knockout: Pruning pre-existing BPE tokenisers with backwards-compatible morphological semi-supervision. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5810–5832, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.324. URL https://aclanthology.org/2024.naacl-long.324/.

Beersmans, M., de Graaf, E., Van de Cruys, T., and Fantoli, M. Training and evaluation of named entity recognition models for classical Latin. In Anderson, A., Gordin, S., Li, B., Liu, Y., and Passarotti, M. C. (eds.), *Proceedings of the Ancient Language Processing Workshop*, pp. 1–12, Varna, Bulgaria, September 2023. INCOMA Ltd., Shoumen, Bulgaria. URL https://aclanthology.org/2023.alp-1.1/.

Bostrom, K. and Durrett, G. Byte pair encoding is suboptimal for language model pretraining. In Cohn, T., He, Y., and Liu, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4617–4624, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.414. URL https://aclanthology.org/2020.findings-emnlp.414/.

Bouma, G. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 30:31–40, 2009.

Boyd, A. and Warmerdam, V. D. floret: lightweight, robust word vectors, Aug 2022. URL https://explosion.ai/blog/floret-vectors.

Burns, P. J. Latincy: Synthetic trained pipelines for latin nlp, 2023. URL https://arxiv.org/abs/2305.04365.

Chang, T. A., Arnett, C., Tu, Z., and Bergen, B. When is multilinguality a curse? language modeling for 250 high- and low-resource languages. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4074–4096, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.236. URL https://aclanthology.org/2024.emnlp-main.236/.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747/.

Cotterell, R., Kirov, C., Sylak-Glassman, J., Walther, G., Vylomova, E., Xia, P., Faruqui, M., Kübler, S., Yarowsky, D., Eisner, J., and Hulden, M. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In Hulden, M. (ed.), *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pp. 1–30, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-2001. URL https://aclanthology.org/K17-2001/.

Creutz, M. and Lagus, K. *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*. Publications in computer and information science. Report A. Helsinki University of Technology, Finland, 2005. ISBN 951-22-7473-6.

Denooz, J. Opera Latina: une base de données sur internet. *Euphrosyne*, 32:79–88, 2004.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Diederich, P. B. Seventeen basic latin endings. *Educational Research Bulletin*, 15(1):1–5, 1936. ISSN 15554023. URL http://www.jstor.org/stable/1471581.

Erdmann, A., Brown, C., Joseph, B., Janse, M., Ajaka, P., Elsner, M., and de Marneffe, M.-C. Challenges and solutions for Latin named entity recognition. In Hinrichs, E., Hinrichs, M., and Trippel, T. (eds.), *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pp. 85–93, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/W16-4012/.

Erdmann, A., Wrisley, D. J., Allen, B., Brown, C., Cohen-Bodénès, S., Elsner, M., Feng, Y., Joseph, B., Joyeux-Prunel, B., and de Marneffe, M.-C. Practical, efficient, and customizable active learning for named entity recognition in the digital humanities. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2223–2234, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1231. URL https://aclanthology.org/N19-1231/.

Erkaya, E. A comprehensive analysis of subword tokenizers for morphologically rich languages. Master's thesis, Boğaziçi University, 2022.

Gamba, F. and Zeman, D. Latin morphology through the centuries: Ensuring consistency for better language processing. In Anderson, A., Gordin, S., Li, B., Liu, Y., and Passarotti, M. C. (eds.), *Proceedings of the Ancient Language Processing Workshop*, pp. 59–67, Varna, Bulgaria, September 2023. INCOMA Ltd., Shoumen, Bulgaria. URL https://aclanthology.org/2023.alp-1.7/.

Ghinassi, I., Tedeschi, S., Marongiu, P., Navigli, R., and McGillivray, B. Language pivoting from parallel corpora for word sense disambiguation of historical languages: A case study on Latin. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N. (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 10073–10084, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.880/.

Gorman, K., McCarthy, A. D., Cotterell, R., Vylomova, E., Silfverberg, M., and Markowska, M. Weird inflects

but OK: Making sense of morphological generation errors. In Bansal, M. and Villavicencio, A. (eds.), *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 140–151, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/K19-1014. URL https://aclanthology.org/K19-1014/.

Gorovaia, S., Schmidt, G., and Yamshchikov, I. P. Sui generis: Large language models for authorship attribution and verification in Latin. In Hämäläinen, M., Öhman, E., Miyagawa, S., Alnajjar, K., and Bizzoni, Y. (eds.), *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, pp. 398–412, Miami, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.nlp4dh-1.39. URL https://aclanthology.org/2024.nlp4dh-1.39/.

Hofmann, V., Pierrehumbert, J., and Schütze, H. Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *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)*, pp. 3594–3608, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.279. URL https://aclanthology.org/2021.acl-long.279/.

Honnibal, M. and Montani, I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

Hsu, J., McDonell, K., Muennighoff, N., Wang, B., Wang, K. A., Zou, A., Gokaslan, A., 2019, V. C., Open-640, Grönroos, S.-A., Virpioja, S., 2020, M. K.-., Morfessor, Hofmann, V., Pierrehumbert, J. B., rich, H.-., Superbizarre, S. ., Schütze, H., Pierre-659, J., Hoogeveen, D., Verspoor, K. M., Ji, Y., Zhou, Z., Liu, H., Lei, T., Joshi, H., Barzilay, R., Jaakkola, K., Tymoshenko, A., Mos-695, Luo, H., Shan, W., Chen, C., and Ding, P. Morphpiece : A linguistic tokenizer for large language models. 2023. URL https://api.semanticscholar.org/CorpusID:266184459.

Kudo, T. Subword regularization: Improving neural network translation models with multiple subword candidates. In Gurevych, I. and Miyao, Y. (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 66–75, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1007. URL https://aclanthology.org/P18-1007/.

Kudo, T. and Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Blanco, E. and Lu, W. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012/.

Lendvai, P. and Wick, C. Finetuning Latin BERT for word sense disambiguation on the thesaurus linguae latinae. In Zock, M., Chersoni, E., Hsu, Y.-Y., and Santus, E. (eds.), *Proceedings of the Workshop on Cognitive Aspects of the Lexicon*, pp. 37–41, Taipei, Taiwan, November 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.cogalex-1.5. URL https://aclanthology.org/2022.cogalex-1.5/.

Libovick'y, J. and Helcl, J. Lexically grounded subword segmentation. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:270620835.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Nzeyimana, A. and Niyongabo Rubungo, A. KinyaBERT: a morphology-aware Kinyarwanda language model. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5347–5363, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.367. URL https://aclanthology.org/2022.acl-long.367/.

Ogueji, K., Zhu, Y., and Lin, J. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In Ataman, D., Birch, A., Conneau, A., Firat, O., Ruder, S., and Sahin, G. G. (eds.), *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pp. 116–126, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.mrl-1.11. URL https://aclanthology.org/2021.mrl-1.11/.

Passarotti, M., Budassi, M., Litta, E., and Ruffolo, P. The lemlat 3.0 package for morphological analysis of Latin. In Bouma, G. and Adesam, Y. (eds.), *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pp. 24–31, Gothenburg, May 2017.

Linköping University Electronic Press. URL https://aclanthology.org/W17-0506/.

Pellegrini, M., Litta, E., Passarotti, M., Mambrini, F., and Moretti, G. The two approaches to word formation in the lila knowledge base of latin resources. In *Proceedings of the third international workshop on resources and tools for derivational morphology (DeriMo 2021)*, pp. 101–109. ATILF & CLLE, 2021.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Riemenschneider, F. and Frank, A. Exploring large language models for classical philology. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15181–15199, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.846. URL https://aclanthology.org/2023.acl-long.846/.

Roelli, P. The corpus corporum, a new open latin text repository and tool. *Archivum Latinitatis Medii Aevi*, 2014. URL https://api.semanticscholar.org/CorpusID:61473619.

Schmidt, C. W., Reddy, V., Zhang, H., Alameddine, A., Uzan, O., Pinter, Y., and Tanner, C. Tokenization is more than compression. In *Conference on Empirical Methods in Natural Language Processing*, 2024. URL https://api.semanticscholar.org/CorpusID:268041516.

Schuster, M. and Nakajima, K. Japanese and korean voice search. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5149–5152, 2012. URL https://api.semanticscholar.org/CorpusID:22320655.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. In Erk, K. and Smith, N. A. (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2020. URL https://arxiv.org/abs/1909.08053.

Springmann, U., Schmid, H., and Najock, D. Latmor: A latin finite-state morphology encoding vowel quantity. *Open Linguistics*, 2, 10 2016. doi: 10.1515/opli-2016-0019.

Sprugnoli, R., Passarotti, M., Cecchini, F. M., Fantoli, M., and Moretti, G. Overview of the EvaLatin 2022 evaluation campaign. In Sprugnoli, R. and Passarotti, M. (eds.), *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pp. 183–188, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lt4hala-1.29/.

Ströbel, P. B. Roberta base latin cased v1, 2022. URL https://huggingface.co/pstroe/roberta-base-latin-cased.

Toraman, C., Yilmaz, E. H., Şahinuç, F., and Ozcelik, O. Impact of tokenization on language models: An analysis for turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22:1 – 21, 2022. URL https://api.semanticscholar.org/CorpusID:248240018.

Vasiu, M. A. and Potolea, R. Enhancing tokenization by embedding romanian language specific morphology. *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 243–250, 2020. URL https://api.semanticscholar.org/CorpusID:227232820.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. Transformers: State-of-the-art natural language processing. In Liu, Q. and Schlangen, D. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

Yehezkel, S. and Pinter, Y. Incorporating context into subword vocabularies. In Vlachos, A. and Augenstein, I. (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 623–635, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.45. URL https://aclanthology.org/2023.eacl-main.45/.

Zouhar, V., Meister, C., Gastaldi, J., Du, L., Sachan, M., and Cotterell, R. Tokenization and the noiseless chan-

nel. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5184–5207, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.284. URL https://aclanthology.org/2023.acl-long.284/.

A. Appendix

A.1. Disambiguating Segmentations with POS Tags

When attempting to match a UD POS tag to a Lemlat POS tag, the Lemlat tags are checked in the order they appear in Table 4.

A.2. Tokenizer Implementation Details

Training Hyperparameters For all tokenizers, we fix the vocabulary size at 30k. For ULM, we set the shrinking factor to the default HuggingFace value, 0.75.

MorphPreTokenization For ULM, we use a sequence of the default `Metaspace()` pretokenizer, followed by a `CharDelimiterSplit(delimiter="@")`. The `Metaspace` pretokenizer replaces whitespace with a special underscore-like symbol, then splits on this character and prepends it to the next word. Functionally, this means that the first subword of a word is differentiated from continuing subwords with this symbol. Then, the `CharDelimiterSplit` pretokenizer will split on our morpheme symbol, allowing morphological suffixes to be treated as continuing subwords.

WordPiece requires more modification than ULM, since the continuing subword symbol "##" is added at train time rather during pretokenization. First, we use a sequence of pretokenizers: `WhitespaceSplit()` followed by `Split(pattern = "@", behavior = "merged_with_next")`. Then, we add a `morph_delimiter="@"` argument to the `WordPieceTrainer`. During training, if a subword is encountered that starts with the `morph_delimiter`, the delimiter is replaced with the continuing subword symbol "##" and the new subword is added to the vocabulary.

A.3. Pretraining Details

Aside from the architecture size and training iterations, we use Megatron-LM's default hyperparameters.

We stop pretraining once reasonable POS accuracy is achieved, which occurred after 500k steps, or about 1.2 epochs. For all 8 models, this took around 715 GPU hours on L40s and A100s.

A.4. Finetuning Details

We finetune for 15 epochs, keeping the model that had the highest whole string morphological accuracy on the validation set.

		Gold = Acontextual Lemlat Segmentations					Downstream Performance	
Tokenizer Type	**Algorithm**	**Exact Match**	**Subword Recall**	**Subword Precision**	**Subword F1**	**Fertility**	**POS Acc**	**Morph Acc**
Baseline	ULM	.1387	.1367	.1371	.1369	1.8714	.9466	.9065
MorphSeed	ULM	.1524	.1576	.1564	.1570	1.8926	.9488	.9024
Acontextual MorphPreTok	ULM	.7001	.8349	.6646	.7401	2.3585	.9581	.9275
Contextual MorphPreTok	ULM	.6544	.8003	.6246	.7016	2.4052	.9598	.9406
Baseline	WP	.2267	.2273	.2412	.2340	1.7688	.9487	.9306
MorphSeed	WP	.2273	.2283	.2423	.2351	1.7688	.9500	.9363
Acontextual MorphPreTok	WP	.8281	.9084	.7993	.8504	2.1335	.9590	.9313
Contextual MorphPreTok	WP	.7538	.8504	.7338	.7878	2.1755	.9623	.9225
		Gold = Contextual Lemlat Segmentations					**Downstream Performance**	
Tokenizer Type	**Algorithm**	**Exact Match**	**Subword Recall**	**Subword Precision**	**Subword F1**	**Fertility**	**POS Acc**	**Morph Acc**
Baseline	ULM	.1076	.1168	.1208	.1188	1.8714	.9466	.9065
MorphSeed	ULM	.1212	.1372	.1404	.1388	1.8926	.9488	.9024
Acontextual MorphPreTok	ULM	.6505	.7762	.6374	.7000	2.3585	.9581	.9275
Contextual MorphPreTok	ULM	.7188	.8498	.6843	.7581	2.4052	.9598	.9406
Baseline	WP	.2012	.2137	.2341	.2234	1.7688	.9487	.9306
MorphSeed	WP	.2018	.2149	.2353	.2246	1.7688	.9500	.9363
Acontextual MorphPreTok	WP	.7550	.8294	.7530	.7893	2.1335	.9590	.9313
Contextual MorphPreTok	WP	.8432	.9190	.8182	.8657	2.1755	.9623	.9225

Table 3. Full Tokenizer Evaluation Metrics

UD POS Tag	**Lemlat POS Tags**
NOUN	Noun, Adjective
PROPN	Noun, Adjective
VERB	Verb
ADJ	Adjective, Noun
PRON	Pronoun, Noun, Invariable
ADV	Invariable
ADP	Preposition, Invariable
CCONJ	Conjunction, Invariable
SCONJ	Conjunction, Invariable
PART	Interjection, Invariable
INTJ	Interjection, Invariable
DET	Pronoun, Adjective
X	Invariable, Other
AUX	Verb
PUNCT	Invariable
NUM	Noun, Adjective, Invariable

Table 4. Mapping from Universal Dependencies (UD) POS Tags to Lemlat POS Tags

Hyperparameter	**Value**
Layers	4
Hidden Size	512
Attention Heads	8
Sequence Length	512
Max Position Embeddings	512
Micro Batch Size	4
Global Batch Size	32
Learning Rate	0.0001
Training Iterations	4,216,370
LR Decay Iterations	990,000
LR Decay Style	Linear
Minimum Learning Rate	1.0×10^{-5}
Weight Decay	0.01
LR Warmup Fraction	0.01
Gradient Clipping	1.0
Mixed Precision (FP16)	True

Table 5. Pretraining Hyperparameters

Hyperparameter	**Value**
Learning Rate	5×10^{-5}
Per Device Train Batch Size	8
Per Device Eval Batch Size	8
Number of Training Epochs	15
Weight Decay	0.01
Evaluation Steps	50

Table 6. Fine-tuning Hyperparameters