MorphBPE: Morphology-Aware Tokenization for Efficient LLM Training

Ehsaneddin Asgari[¶,1], **Yassine El Kheir**[1,2*], **Mohammad Ali Sadraei Javaheri**[1], **Ali Nazari**[1]
[1] Qatar Computing Research Institute (QCRI), Hamad Bin Khalifa University (HBKU), Doha, Qatar
[2] German Research Center for Artificial Intelligence (DFKI) / Technical University of Berlin, Germany
[¶] Corresponding Author: easgari@hbku.edu.qa

Abstract

Tokenization fundamentally shapes NLP performance, affecting both efficiency and linguistic fidelity. While Byte Pair Encoding (BPE) underpins most Large Language Models (LLMs), its frequency-driven merges often disregard morpheme boundaries, yielding inconsistent and semantically opaque segmentations in morphologically rich languages. We introduce MorphBPE, a simple extension of BPE that constrains merge operations during tokenizer training to respect morpheme boundaries, while leaving inference unchanged and fully compatible with existing LLM pipelines. We evaluate tokenization quality using two intrinsic metrics: Morphological Consistency F1, which measures whether shared morphemes are assigned consistent token representations, and Morphological Edit Distance, which quantifies alignment with morpheme boundaries. We then train 300M and 1B parameter decoder-only LMs from scratch across four typologically diverse languages: English, Russian, Hungarian, and Arabic, under identical vocabulary sizes and training settings. Across all languages, MorphBPE consistently improves intrinsic morphological coherence and reduces language model cross-entropy; moreover, token length statistics indicate that these gains are not attributable to materially shorter tokens. Finally, on the Belebele multilingual reading comprehension benchmark, MorphBPE yields significant improvements in morphologically rich languages such as Russian and Arabic.

Availability: The *MorphBPE* codebase, datasets, and tokenizers are available at: https://github.com/qcri/MorphBPE.

1 Introduction

Tokenization is a central design choice in modern NLP systems and a critical bottleneck for multilingual Large Language Models (LLMs). By mapping raw text into discrete units such as bytes (Gillick et al., 2016), characters (Al-Rfou et al., 2019), subwords (Sennrich et al., 2016), or words, tokenization directly determines vocabulary size, sequence length, and the granularity at which linguistic regularities can be learned. Errors or inconsistencies introduced at this stage propagate through the entire modeling pipeline and can substantially affect both training efficiency and downstream performance (Sajjad et al., 2017; Adel et al., 2018). Despite growing interest in tokenization-free or character-level alternatives (Clark et al., 2022; Deiseroth et al., 2024), nearly all state-of-the-art LLMs, including Gemma (Team et al., 2024), LLaMA (Touvron et al., 2023), DeepSeek (Bi et al., 2024), and OpenAI's GPT family, rely on Byte Pair Encoding (BPE) or closely related variants due to their favorable trade-offs between efficiency and coverage.

BPE is a frequency-driven algorithm that iteratively merges common symbol pairs, making it well suited for concatenative morphology, as in English, where morphemes are typically formed by linear affixation. However, this same mechanism leads to systematic failures in languages with richer or more complex morphological systems. In non-concatenative languages such as Arabic and Hebrew, meaning is expressed through root-and-pattern morphology rather than linear affixation, and frequent substrings do not necessarily correspond to meaningful units (Khaliq and Carroll, 2013). Agglutinative languages such as Hungarian, Turkish, and Korean further challenge BPE, as long sequences of productive affixes result in a large space of related word forms that BPE fragments inconsistently (Hakkani-Tür et al., 2000). As a result, standard BPE tokenizations often fail to align with true morpheme boundaries, producing subword units that are neither linguistically interpretable nor stable across related word forms. In practice, BPE segmentations in morphologically rich languages frequently introduce semantic ambiguity by reusing frequent substrings across unrelated words. For example, in Arabic the word الرحمن (Al-Rahman, "The Merciful") may be segmented into من (min, "whom"), ال (al, "the"), and

[*] Work conducted while at QCRI.

Findings of the Association for Computational Linguistics: ACL 2026, pages 41610–41621

رح, even though من is semantically unrelated to the original word. Such segmentations force the model to disentangle spurious token-level associations during training, increasing the burden on representation learning. Similar phenomena arise in agglutinative languages, where morphemes expressing tense, number, or case are split inconsistently across tokens, undermining the model's ability to generalize across inflected forms.

A natural solution is to incorporate morphological information into tokenization. However, purely morphology-driven segmentation has been shown to conflict with corpus statistics and can lead to inefficient vocabularies or brittle behavior when applied naively (Durrani et al., 2019; Marco and Fraser, 2024). This highlights a key open challenge for multilingual NLP: how to enforce morphological coherence in tokenization without sacrificing the statistical efficiency and scalability that make BPE attractive for large-scale LLM training.

In this work, we introduce $MorphBPE$, a simple and practical extension of BPE that constrains merge operations during tokenizer training to respect morpheme boundaries. Unlike purely morphology-based segmentation, MorphBPE remains fully data-driven, allowing frequent and productive morphemes to be preferentially captured while avoiding unnecessary fragmentation of rare or uninformative morphemes. The constraint is applied only during training, and the resulting tokenizer behaves identically to standard BPE at inference, introducing no additional runtime cost and remaining fully compatible with existing LLM architectures and deployment pipelines. A central motivation for MorphBPE is *morphological consistency*. Beyond aligning tokens with morpheme boundaries, consistency requires that words sharing morphemes are represented using shared subword units, and that shared tokens correspond to shared morphological content. By integrating morphological constraints into a frequency-based merge objective, MorphBPE yields more stable and interpretable subword representations, facilitating more efficient language model training and improved performance, particularly in morphologically rich languages.

Contributions: **(i)** We propose $MorphBPE$, a morphology-aware extension of BPE that constrains merge operations using morpheme boundaries during tokenizer training, while leaving inference unchanged and fully compatible with existing LLM pipelines. **(ii)** We introduce two intrinsic metrics for evaluating morphological quality of tokenizers: **Morphological Consistency F1**, which quantifies consistency across words sharing morphemes, and **Morphological Edit Distance**, which measures alignment with morpheme boundaries. **(iii)** We conduct controlled language model experiments with 300M and 1B parameter models trained from scratch across four typologically diverse languages, English, Russian, Hungarian, and Arabic, showing that MorphBPE consistently reduces cross-entropy under identical vocabulary sizes. **(iv)** We demonstrate that these intrinsic improvements translate into measurable gains on the Belebele multilingual reading comprehension benchmark, with statistically significant improvements in morphologically rich languages such as Russian and Arabic.

By integrating linguistic principles with modern tokenization strategies, MorphBPE bridges the gap between traditional morphological analysis and NLP, providing a computationally efficient and morphologically interpretable tokenization approach for language modeling, particularly in morphologically rich languages like Arabic. In line with this, MorphBPE has been developed and implemented in Fanar (Team et al., 2025; FANAR TEAM et al., 2026), an Arabic-centric language model, leading to significant improvements in model performance.

2 Background and Related Work

Subword Tokenization in Language Models: Subword tokenization has become a foundational component of modern neural language models, enabling a balance between open-vocabulary coverage and computational efficiency. Early approaches explored character-level modeling (Al-Rfou et al., 2019) and byte-level representations (Gillick et al., 2016), which offer robustness across scripts but often incur longer sequences and higher computational cost. Subword-based methods, most notably Byte Pair Encoding (BPE) (Sennrich et al., 2016), WordPiece, and the SentencePiece Unigram LM (Kudo and Richardson, 2018), emerged as practical compromises, allowing frequent patterns to be captured while decomposing rare words into reusable units.

BPE, originally proposed as a text compression algorithm (Shibata et al., 1999), was adapted for neural machine translation in 2016 and rapidly

became the de facto standard in NLP and Large Language Models (LLMs). Its popularity stems from its simplicity, deterministic behavior, and effectiveness in controlling vocabulary size while handling out-of-vocabulary words. Numerous extensions have been proposed to mitigate its shortcomings, including BPE-Dropout (Provilkov et al., 2020), which injects stochasticity to improve generalization, sampling-based and probabilistic variants (Asgari et al., 2020), byte-level adaptations for improved robustness across scripts (Wang et al., 2020), and multilingual BPE schemes designed to encourage cross-lingual token sharing (Liang et al., 2023). Despite these advances, most BPE-based approaches remain purely frequency-driven and largely agnostic to linguistic structure.

Morphology-Aware Tokenization: The limitations of frequency-based tokenization in morphologically rich languages have motivated a growing body of work on morphology-aware tokenization. Early approaches often rely on explicit morphological analyzers or pre-tokenization strategies, segmenting words into morphemes prior to applying a standard subword tokenizer (Otani et al., 2020; Nzeyimana and Niyongabo Rubungo, 2022). Other methods incorporate morphological dictionaries or multi-view segmentation signals (Park et al., 2021), aiming to expose models to linguistically meaningful units.

More recent work has explored hybrid approaches that seek to balance morphological structure with statistical efficiency. For example, morpheme-aware or linguistically informed tokenizers encourage alignment between subwords and morpheme boundaries (Jabbar, 2023; Marco and Fraser, 2024), while analytical studies examine how subword segmentation interacts with morphological complexity and model performance (Weller-Di Marco and Fraser, 2024). However, many of these methods introduce additional preprocessing stages, require runtime morphological analysis, rely on stochastic objectives, or depart substantially from standard BPE training and inference pipelines. As a result, their adoption in large-scale LLM training remains limited.

Positioning of MorphBPE: MorphBPE is designed to address these limitations while preserving the practical advantages that have made BPE ubiquitous. Importantly, MorphBPE is *not* a runtime morphological analyzer, nor does it require morphological lookup or rule-based processing at inference time. All morphological information is used solely during tokenizer training to constrain merge operations, and the resulting tokenizer behaves identically to standard BPE at inference. What distinguishes MorphBPE from prior work is the point of integration and the simplicity of the constraint. Rather than pre-segmenting text (pre-tokenization) or post-processing tokenizations, MorphBPE injects morphological structure directly into the BPE merge objective by disallowing merges that cross known morpheme boundaries. This yields a deterministic tokenizer, avoids the memory and sampling overhead of probabilistic approaches such as the Unigram LM, and maintains full compatibility with existing LLM architectures and training workflows. MorphBPE bridges the linguistic motivation and the data-driven thinking in a way that prior approaches have not fully achieved.

3 MorphBPE Method

MorphBPE is guided by four core design principles motivated by practical deployment requirements for large-scale multilingual LLMs. **(i) Minimal intervention:** Rather than redesigning tokenization from scratch, MorphBPE introduces a single, targeted modification to the standard BPE training procedure. The goal is to preserve the empirical strengths of BPE while correcting its systematic failures on morphological structure. **(ii) Determinism:** MorphBPE is fully deterministic. Unlike stochastic tokenization schemes such as BPE-Dropout or Unigram LM sampling, identical inputs and hyperparameters always produce the same tokenizer. This property is important for reproducibility and large-scale training stability. **(iii) Training-time supervision only:** Morphological information is used exclusively during tokenizer training to guide merge decisions. At inference time, MorphBPE behaves identically to standard BPE and does not require morphological annotation, lookup tables, or runtime analysis. **(iv) Pipeline compatibility:** The resulting tokenizer is a standard BPE model and integrates seamlessly into existing LLM training and inference pipelines without architectural or infrastructural changes.

3.1 Algorithm

Byte Pair Encoding (BPE) begins with a character-level vocabulary and iteratively merges the most frequent adjacent symbol pairs until a target vocabulary size is reached (Sennrich et al., 2016). Merge decisions are purely frequency-based and

Algorithm 1 Morphology-Aware Byte Pair Encoding (MorphBPE)

```
Initialize vocabulary with individual characters
Obtain morpheme boundaries for the training corpus
while number of merges < target vocabulary size do
    Compute frequencies of adjacent symbol pairs
    Select the most frequent pair that does not cross a morpheme boundary
    Merge the selected pair and update the vocabulary
end while
```

unconstrained by linguistic structure.

MorphBPE modifies this process by introducing a single constraint. During tokenizer training, merges that cross known morpheme boundaries are disallowed. All other aspects of BPE, including frequency computation, merge ranking, and inference behavior, remain unchanged. Formally, let a word be segmented into a sequence of morphemes according to available morphological annotations. During BPE training, a candidate merge between symbols x and y is permitted only if both symbols belong to the same morpheme span. If the merge would combine symbols originating from different morphemes, it is skipped in favor of the next most frequent valid pair.

Algorithm 1 summarizes the procedure.

Because constraints are applied only during merge selection, the learned merges define a standard BPE tokenizer. No additional metadata or morphology-specific logic is required at inference.

3.2 Practical Properties

MorphBPE has the same asymptotic time complexity as standard BPE. The additional cost of checking morpheme boundary constraints is linear in the number of candidate merges and negligible relative to frequency computation. Memory usage is unchanged, as morpheme boundaries are required only during tokenizer training and are not stored in the final tokenizer.

From a systems perspective, MorphBPE is drop-in compatible with existing tokenization libraries and LLM training frameworks. Models trained with MorphBPE require no changes to architectures, batching strategies, or inference code. This makes MorphBPE suitable for large-scale training regimes where reproducibility, efficiency, and simplicity are critical.

4 Evaluation Framework

We evaluate MorphBPE at three complementary levels in order to isolate the effects of morphology-aware tokenization and avoid confounding factors. First, we assess intrinsic properties of the tokenizer itself, independent of any language model. Second, we evaluate intrinsic language model behavior during training using controlled cross-entropy comparisons. Finally, we measure downstream task performance to determine whether intrinsic improvements translate into practical gains. This layered evaluation design makes the logic of our experiments explicit and aligns with best practices for analyzing tokenization methods in LLMs.

4.1 Tokenizer-Level Evaluation (Intrinsic)

Tokenizer-level evaluation focuses on properties of the segmentation itself, without involving language model training. These metrics directly quantify how well a tokenizer aligns with morphological structure and how efficiently it represents text.

Fertility: Fertility (ϕ) measures the average number of subword tokens produced per word relative to a whitespace-based baseline (Rust et al., 2021). Lower fertility indicates higher compression efficiency and potentially longer effective context windows. However, fertility must be interpreted cautiously, as morphologically rich languages such as Hungarian and Arabic naturally require higher fertility to encode productive inflectional and derivational processes. We therefore report fertility alongside morphology-sensitive metrics rather than treating it as a standalone indicator of quality.

Morphological Edit Distance: We introduce Morphological Edit Distance (μ_e), an intrinsic metric that quantifies alignment between tokenizer output and gold morpheme boundaries. Using a dynamic programming alignment that preserves token order, this metric measures the minimum number of insertions, deletions, and substitutions required to transform a token sequence into a morpheme sequence. Lower values indicate better adherence to morphological structure and greater interpretability. We report raw edit distances to reflect the average number of boundary mismatches.

Morphological Consistency F1: Morphological Consistency F1 (μ_c), inspired by Marco and Fraser (2024), measures whether segmentation decisions are consistent across words that share morphemes. Recall captures whether words with shared morphemes receive shared tokens, while precision measures whether shared tokens correspond to shared morphemes. Their harmonic mean yields μ_c. To ensure scalability, we cluster words using k-means ($k = 100$), sample $C = 50$ word pairs per clus-

ter, and estimate scores with $N = 10$ bootstrap resamples.

4.2 Language Model Evaluation (Intrinsic)

Intrinsic language model evaluation assesses how tokenization affects training dynamics and representational efficiency when all other factors are held constant.

Token-Level Cross-Entropy. We train decoder-only language models from scratch using either BPE or MorphBPE and compare token-level cross-entropy loss during training. Cross-entropy reflects both convergence speed and the quality of learned representations and is closely related to perplexity while providing finer-grained resolution. Comparisons are performed only between models with identical architectures, training data, and vocabulary sizes to ensure fairness.

Rationale for Fair Comparison. Vocabulary size directly affects branching factors and loss values, making unequal vocabularies incomparable. We therefore fix vocabulary sizes per language and show empirically that MorphBPE and BPE produce nearly identical token-length distributions, ruling out explanations based on trivial token shortening (see Appendix B for detailed analysis). Under these controlled conditions, differences in cross-entropy can be attributed to segmentation quality rather than capacity or compression artifacts.

4.3 Downstream Evaluation (Extrinsic)

Extrinsic evaluation measures whether intrinsic improvements translate into gains on real language understanding tasks. We use the **LM Evaluation Harness** (Gao et al., 2024) to conduct evaluations under a standardized zero-shot protocol. We evaluate models on the Belebele multilingual reading comprehension benchmark (Bandarkar et al., 2024), which consists of multiple-choice questions derived from Flores-200 passages across 122 language variants. We report results for English, Russian, Hungarian, and Arabic, enabling direct comparison across morphological typologies.

Evaluation Protocol and Significance Testing: For each example, models score candidate answers by conditional log-probability and select the most likely option. Accuracy is computed over all examples. To assess statistical significance, we apply McNemar's test on paired predictions, combined with 10^6 bootstrap resamples of accuracy differences. Multiple comparisons are corrected using the Benjamini–Hochberg false discovery rate procedure with $\alpha = 0.05$. This protocol allows us to determine whether observed gains are robust rather than due to sampling noise.

5 Experimental Setup

This section details the data resources, tokenizer training procedure, and language model training configuration used in our experiments. The goal is to ensure reproducibility and make explicit the controls used to enable fair comparisons between standard BPE and MorphBPE.

5.1 Morphological Resources

Morphological supervision is required only during tokenizer training. We use manually annotated and high-confidence automatically generated morpheme segmentations covering four typologically diverse languages: English, Russian, Hungarian, and Arabic.

For English, Russian, and Hungarian, morphological segmentations are obtained from the SIGMORPHON 2022 Shared Task on Morpheme Segmentation (Batsuren et al., 2022). These datasets provide high-quality gold annotations covering both inflectional and derivational morphology. For Arabic, which exhibits non-concatenative templatic morphology, we combine multiple complementary resources: the Arabic Treebank (ATB) (Taji et al., 2017), the Dialectal Segmentation Dataset (Darwish et al., 2018), and the Quranic Morphology dataset (Dukes and Habash, 2010). To increase coverage of frequent surface forms, we additionally include one million high-confidence Arabic segmentations generated using Farasa (Darwish and Mubarak, 2016).

All datasets are normalized to a unified segmentation format and deduplicated. Manually annotated resources are split into 80% training, 10% validation, and 10% test sets. Automatically generated Arabic segmentations are used only for tokenizer training and excluded from intrinsic evaluation. Dataset statistics are summarized in Table 1.

5.2 LLM Training Data

For language model training, we use the FineWeb2 corpus (Penedo et al., 2024), a large-scale multilingual web dataset covering over 1,000 languages. FineWeb2 provides sufficient data volume and linguistic diversity to support controlled monolingual training while adhering to the Chinchilla scaling law (Hoffmann et al., 2022).

Table 1: Morphological segmentation dataset statistics used for BPE and $MorphBPE$ training and evaluation across languages.

Language	Morphology Type	# of Words	Avg. Morphemes per Word
English	Fusional (low complexity)	571,495	2.33
Russian	Fusional (moderate complexity)	784,212	3.84
Hungarian	Agglutinative (high complexity)	930,312	3.22
Arabic	Templatic (high complexity)	1,395,835	2.50

For each language, we extract language-specific subsets using FineWeb2 metadata and train models on a fixed token budget. This ensures that differences in model behavior arise from tokenization rather than data scale. Using the same underlying text for both BPE and MorphBPE further isolates the effect of segmentation.

5.3 Tokenizer Training Setup

Both BPE and MorphBPE tokenizers are trained using identical corpora and target vocabulary sizes. MorphBPE differs only in that morpheme boundary constraints are applied during merge selection, as described in Section 3. All other aspects of tokenizer training, including initialization, frequency computation, and merge ordering, remain unchanged.

Vocabulary sizes are selected separately for each language following a principled procedure. We train tokenizers with vocabulary sizes ranging from 8K to 96K in 8K increments and compute morphological edit distance on development sets. The smallest vocabulary size beyond which improvements are not statistically significant is selected using a paired t-test. This yields 24K for Hungarian, 64K for Russian, and 96K for English and Arabic. These sizes are then fixed for all tokenizer-level and language model experiments to ensure fair comparison.

5.4 Language Model Training Setup

We train decoder-only Transformer language models at two scales: a 300M-parameter model and a 1B-parameter model. All models use identical architectures, optimization settings, and training schedules, and differ only in the tokenizer used. Training is implemented using the LLaMA-Factory framework (Zheng et al., 2024).

The 300M models are trained on approximately 6B tokens, while the 1B models are trained on approximately 20B tokens, consistent with Chinchilla-optimal scaling. Optimization uses AdamW with cosine learning rate decay, identical batch sizes, and the same random seeds across tokenizer variants. All experiments are conducted on H100 GPUs, with total compute on the order of several thousand GPU-hours.

By tightly controlling model architecture, data, vocabulary size, and training budget, this setup ensures that any observed differences in training dynamics or downstream performance can be attributed to the tokenizer rather than confounding factors.

6 Results

6.1 Tokenizer-Level Results

Figure 1 and Table 2 summarize intrinsic tokenizer results across English, Russian, Hungarian, and Arabic. Across all languages, MorphBPE achieves consistently lower Morphological Edit Distance (μ_e) and higher Morphological Consistency F1 (μ_c) than standard BPE, while incurring only a marginal increase in fertility (ϕ). The improvements are largest for morphologically rich languages, especially Hungarian and Arabic, where respecting morpheme boundaries prevents frequent but linguistically spurious merges.

For completeness, we also compare Morphological Consistency F1 (μ_c) against SentencePiece Unigram LM (Kudo and Richardson, 2018) under identical vocabulary sizes and corpora (Table 2). MorphBPE achieves higher μ_c across all four languages, indicating more stable and morphologically coherent tokenization.

Overall, these results confirm that constraining merges at morpheme boundaries produces tokenizations that better preserve morphological structure and reduce segmentation ambiguity. This effect increases with morphological complexity, supporting the motivation for morphology-aware tokenization in agglutinative and templatic languages.

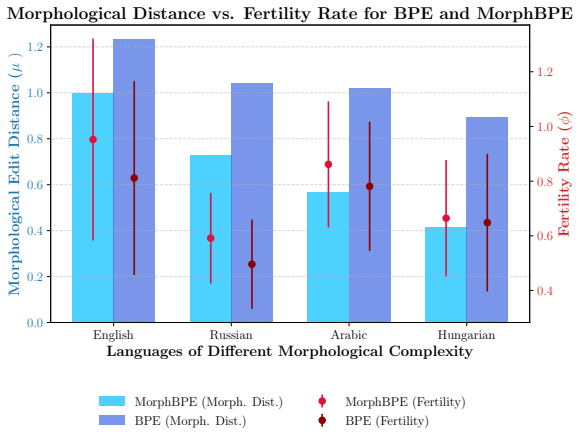

Figure 1: Comparison of morphological distance and fertility rate for BPE and $MorphBPE$ across four languages. Lower fertility is generally preferred, and lower morphological distance indicates better alignment with morpheme boundaries.

6.2 Language Model Results (Intrinsic)

Figure 2 reports training cross-entropy curves for the 300M and 1B models across the four languages. Under identical vocabulary sizes, architectures, and training data, MorphBPE consistently yields lower cross-entropy than BPE. The trend is stable across both scales and persists throughout training, with Figure 2 showing a representative window of approximately 14B tokens for readability.

The cross-entropy reductions indicate more efficient learning dynamics when token boundaries align with morphological structure. Improvements are most pronounced for morphologically rich languages such as Hungarian and Arabic, where MorphBPE provides clearer subword regularities and reduces the burden of learning morphology from fragmented and inconsistent substrings. Importantly, these gains arise under controlled settings, suggesting they are attributable to segmentation quality rather than vocabulary capacity or data scale.

6.3 Downstream Results (Extrinsic)

We evaluate downstream effects using the Belebele multilingual reading comprehension benchmark (Bandarkar et al., 2024). Because our goal is to isolate tokenization effects, we report zero-shot performance for the 300M and 1B models without supervised fine-tuning. Absolute accuracy is therefore modest, but relative differences remain informative and can be assessed statistically.

Across the four languages, MorphBPE yields an average accuracy gain of approximately 1% over BPE. Gains are consistent for morphologically rich languages, Arabic, Russian, and Hungarian, and negligible for English, where morphology is less productive. Using McNemar's paired test with 10^6 bootstrap resamples and Benjamini–Hochberg correction at $\alpha = 0.05$, improvements for Arabic and Russian are statistically significant, while the Hungarian gain is not.

These results suggest that morphology-aware tokenization can translate into measurable downstream benefits for morphologically complex languages, complementing intrinsic tokenizer metrics and cross-entropy improvements. Together, the intrinsic and extrinsic results support the claim that improving morphological alignment at the tokenization stage strengthens representation learning and downstream comprehension.

7 Analysis and Discussion

When and Why MorphBPE Helps: MorphBPE is most effective in languages with rich and productive morphology, where surface word forms encode substantial grammatical and semantic information through affixation or non-concatenative processes. In agglutinative languages such as Hungarian and templatic languages such as Arabic, standard BPE frequently fragments words into statistically frequent but linguistically incoherent substrings. By constraining merges to respect morpheme boundaries, MorphBPE produces more interpretable and stable subword units that align with meaningful linguistic structure. This improved alignment manifests in higher morphological consistency and lower morphological edit distance, indicating that related word forms are segmented in a more systematic manner. From a representation learning perspective, such consistency reduces ambiguity in token semantics and allows the language model to reuse parameters across morphologically related forms more effectively. The resulting representations are therefore easier to learn and generalize, which is reflected in faster convergence and lower cross-entropy loss during training. In languages with relatively simple or weak morphology, such as English, the benefits of morphology-aware tokenization are naturally limited. English word formation relies less on productive inflection and more on fixed lexical items, and standard BPE already performs reasonably well at capturing frequent subword patterns. As a result, MorphBPE yields only

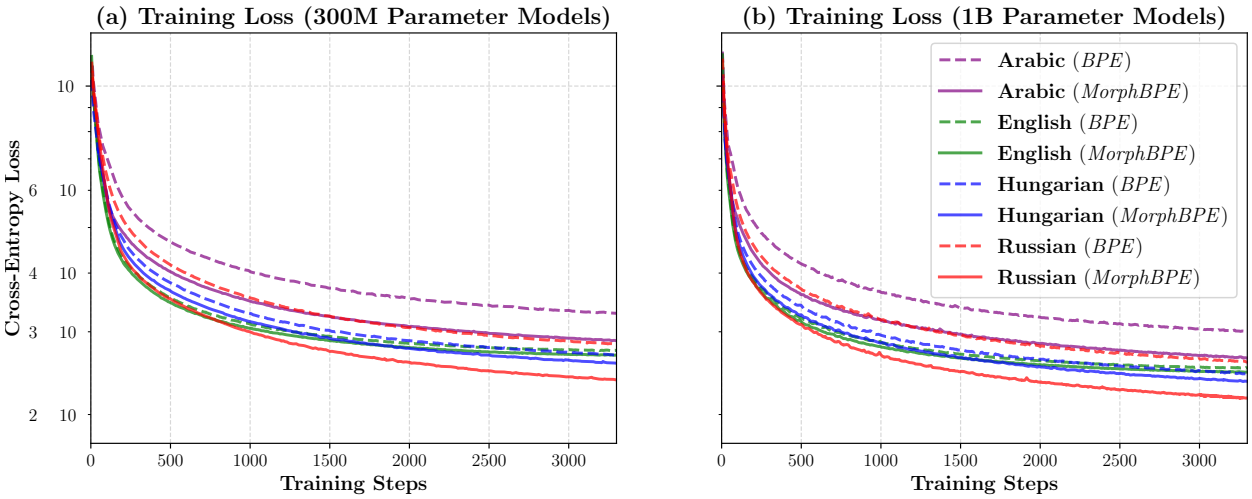

Figure 2: Training cross-entropy loss comparison between BPE and $MorphBPE$ across English, Russian, Hungarian, and Arabic for both the 300M and 1B models (lower values indicate better performance).

modest improvements in intrinsic tokenizer metrics and negligible gains in downstream accuracy for English.

These findings are expected and highlight an important property of MorphBPE. The method does not degrade performance in low-morphology settings, but its advantages emerge primarily when morphological structure plays a central role in word formation. This behavior suggests that MorphBPE is a targeted improvement rather than a universally transformative change to tokenization.

Table 2: Morphological consistency evaluation for BPE, $MorphBPE$, and SentencePiece Unigram LM across four languages. Precision, recall, and F1-score (μ_c) are reported as mean $\pm$ standard deviation over multiple resamples of the test sets. Higher F1-scores indicate greater consistency in segmenting words that share or differ in morphemes.

Model	Precision (Mean $\pm$ Std)	Recall (Mean $\pm$ Std)	Morph. F1 (μ_c)
English (96K)			
BPE	0.00 ± 0.00	0.03 ± 0.02	0.00
MorphBPE	0.31 ± 0.44	0.34 ± 0.09	**0.32**
SentencePiece Unigram LM	0.20 ± 0.41	0.44 ± 0.07	0.28
Russian (64K)			
BPE	0.10 ± 0.32	0.06 ± 0.01	0.07
MorphBPE	0.69 ± 0.48	0.33 ± 0.06	**0.45**
SentencePiece Unigram LM	0.63 ± 0.49	0.22 ± 0.06	0.33
Hungarian (24K)			
BPE	0.08 ± 0.25	0.29 ± 0.04	0.13
MorphBPE	0.98 ± 0.03	0.78 ± 0.07	**0.87**
SentencePiece Unigram LM	0.93 ± 0.17	0.81 ± 0.10	0.87
Arabic (96K)			
BPE	0.00 ± 0.00	0.08 ± 0.03	0.00
MorphBPE	0.89 ± 0.31	0.53 ± 0.05	**0.66**
SentencePiece Unigram LM	0.73 ± 0.27	0.49 ± 0.04	0.58

Tokenization, Fertility, and Vocabulary Size: A recurring assumption in tokenizer evaluation is that lower fertility or shorter token sequences directly indicate better tokenization. Our results challenge this view. While MorphBPE sometimes produces slightly higher fertility than BPE, especially in morphologically rich languages, it consistently yields better morphological alignment and improved language model performance.

These findings indicate that fertility alone is an insufficient proxy for tokenizer quality. Vocabulary size and compression interact with linguistic structure in complex ways, and aggressive compression can obscure systematic morphological patterns that are beneficial for learning. MorphBPE demonstrates that modest increases in token count can be offset by gains in interpretability, consistency, and representation stability, leading to more efficient learning overall.

8 Conclusion

We introduced MorphBPE, a morphology-aware extension of Byte Pair Encoding that integrates linguistic structure into subword tokenization while preserving the efficiency, determinism, and compatibility of standard BPE. Across four typologically diverse languages, MorphBPE consistently improves morphological alignment, tokenizer consistency, and language model training dynamics, with downstream gains for morphologically rich languages.

Our results demonstrate that respecting morpheme boundaries during tokenizer training leads to more stable and interpretable representations, challenging the view that compression alone determines tokenizer quality. MorphBPE provides a

practical and scalable way to incorporate linguistic insight into LLM training pipelines, and we hope it encourages further exploration of morphology-aware methods for multilingual language modeling.

9 Limitations

While MorphBPE offers significant advantages for morphologically rich languages, we acknowledge certain limitations that frame directions for future research: First, MorphBPE requires access to morphological segmentation data during the tokenizer training phase. It is important to clarify that this does not involve inference-time lookup tables, morphological lexicons, or rule-based systems. Instead, morphology-derived boundaries guide BPE merge decisions during training, resulting in a fully data-driven, standard BPE tokenizer. While high-quality resources exist for many morphologically rich languages (e.g., via UniMorph, SIGMORPHON, and MorphyNet, covering $\sim$100 languages), coverage remains limited for some low-resource and under-documented languages. However, the set of languages with reliable segmentation resources often aligns with those possessing sufficient corpora for meaningful LLM training.

Second, our current experiments focus on monolingual models to isolate the effects of tokenization across distinct typologies. Extending MorphBPE to multilingual models with joint vocabularies introduces new challenges, particularly in balancing morphological representation across diverse language families within a shared subword space. Investigating the interaction between morphology-aware constraints and cross-lingual vocabulary sharing is an open question.

Finally, our current study involved training 16 models from scratch across four languages and two scales (300M and 1B parameters), utilizing approximately 2,000 H100 GPU-hours. This represents a substantial effort to ensure rigorous, controlled comparison. While scaling further to larger model sizes and broader downstream tasks like instruction following or reasoning would require industrial-scale infrastructure, our findings are consistent with established links between improved perplexity and downstream performance (Wei et al., 2024), suggesting that the benefits of MorphBPE are likely to scale.

Acknowledgments

We extend our gratitude to Sanjay Chawla, Mohamed Eltabakh, Ahmed Ali, Muhammad Tasnim Mohiuddin, Sabri Boughorbel, Hamdy S. Mubarak, MohammadAmin Sadeghi, Natasa Milic-Frayling, Nadir Durrani, Mohsen Mahdavi Mazdeh, and the entire Fanar team for their valuable feedback and insights.

Acknowledgment of AI-Assisted Proofreading Claude[1], ChatGPT[2], and Gemini[3] were used solely for proofreading and language refinement. These tools assisted with grammar, spelling, clarity, and sentence structure. All substantive content, ideas, analysis, and conclusions are the author's own.

References

Heike Adel, Ehsaneddin Asgari, and Hinrich Schütze. 2018. Overview of character-based models for natural language processing. In *Computational Linguistics and Intelligent Text Processing: 18th International Conference, CICLing 2017, Budapest, Hungary, April 17–23, 2017, Revised Selected Papers, Part I 18*, pages 3–16, Cham. Springer International Publishing.

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3159–3166.

Ehsaneddin Asgari, Masoud Jalili Sabet, Philipp Dufter, Christopher Ringlstetter, and Hinrich Schütze. 2020. Subword sampling for low resource word alignment. *arXiv preprint arXiv:2012.11657*.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand. Association for Computational Linguistics.

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

[1] Anthropic, Claude
[2] OpenAI, ChatGPT
[3] Google, Gemini

SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Jonathan H Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Kareem Darwish and Hamdy Mubarak. 2016. Farasa: A new fast and accurate Arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1070–1074, Portorož, Slovenia. European Language Resources Association (ELRA).

Kareem Darwish, Hamdy Mubarak, Ahmed Abdelali, Mohamed Eldesouki, Younes Samih, Randah Alharbi, Mohammed Attia, Walid Magdy, and Laura Kallmeyer. 2018. Multi-dialect Arabic POS tagging: A CRF approach. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Björn Deiseroth, Manuel Brack, Patrick Schramowski, Kristian Kersting, and Samuel Weinbach. 2024. T-FREE: Subword tokenizer-free generative LLMs via sparse representations for memory-efficient embeddings. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21829–21851, Miami, Florida, USA. Association for Computational Linguistics.

Kais Dukes and Nizar Habash. 2010. Morphological annotation of Quranic Arabic. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. In *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)*, pages 1504–1516, Minneapolis, Minnesota. Association for Computational Linguistics.

FANAR TEAM, Ummar Abbas, Mohammad Shahmeer Ahmad, Minhaj Ahmad, Abdulaziz Al-Homaid, Anas Al-Nuaimi, Enes Altinisik, Ehsaneddin Asgari, Sanjay Chawla, Shammur Chowdhury, et al. 2026. Fanar 2.0: Arabic generative ai stack. *arXiv preprint arXiv:2603.16397*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. The language model evaluation harness.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1296–1306, San Diego, California. Association for Computational Linguistics.

Dilek Z. Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2000. Statistical morphological disambiguation for agglutinative languages. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*.

J. Hoffmann, S. Borgeaud, M. Arthur, E. Buchatskaya, T. Cai, E. Rutherford, D. Casas, L. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. Driessche, B. Damoc, A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. Rae, O. Vinyals, and L. Sifre. 2022. training compute-optimal large language models. *Arxiv*.

Haris Jabbar. 2023. Morphpiece: Moving away from statistical language representation. *arXiv preprint arXiv:2307.07262*.

Bilal Khaliq and John Carroll. 2013. Induction of root and pattern lexicon for unsupervised morphological analysis of Arabic. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1012–1016, Nagoya, Japan. Asian Federation of Natural Language Processing.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore. Association for Computational Linguistics.

Marion Di Marco and Alexander Fraser. 2024. Subword segmentation in LLMs: Looking at inflection and consistency. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12050–12060, Miami, Florida, USA. Association for Computational Linguistics.

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

Naoki Otani, Satoru Ozaki, Xingyuan Zhao, Yucen Li, Micaelah St Johns, and Lori Levin. 2020. Pretokenization of multi-word expressions in cross-lingual word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4451–4464, Online. Association for Computational Linguistics.

Hyunji Hayley Park, Katherine J. Zhang, Coleman Haley, Kenneth Steimel, Han Liu, and Lane Schwartz. 2021. Morphology matters: A multilingual language modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Martin Jaggi, Leandro von Werra, and Thomas Wolf. 2024. Fineweb2: A sparkling update with 1000s of languages. *HuggingFace*.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, Ahmed Abdelali, Yonatan Belinkov, and Stephan Vogel. 2017. Challenging language-dependent segmentation for Arabic: An application to machine translation and part-of-speech tagging. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 601–607, Vancouver, Canada. Association for Computational Linguistics.

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.

Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. 1999. Byte pair encoding: A text compression scheme that accelerates pattern matching. *Technical Report DOI-TR-161, Department of Informatics, Kyushu University*.

Dima Taji, Nizar Habash, and Daniel Zeman. 2017. Universal Dependencies for Arabic. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 166–176, Valencia, Spain. Association for Computational Linguistics.

Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, et al. 2025. Fanar: An arabic-centric multimodal generative ai platform. *arXiv preprint arXiv:2501.13944*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9154–9160.

Chengwei Wei, Yun-Cheng Wang, Bin Wang, C-C Jay Kuo, et al. 2024. An overview of language models: Recent developments and outlook. *APSIPA Transactions on Signal and Information Processing*, 13(2).

Marion Weller-Di Marco and Alexander Fraser. 2024. Analyzing the understanding of morphologically complex words in large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1009–1020, Torino, Italia. ELRA and ICCL.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Impact of Training Data Segmentation (Full-Text Sanity Check)

In our main experiments across four languages, tokenizers were trained on word lists to ensure highly controlled comparisons. To verify that our findings hold when tokenizers are trained on full running

text, we conducted an additional sanity check using the full Arabic Wikipedia corpus.

In this experiment, we compare two approaches trained directly on the full corpus:

1. **Standard BPE (Raw Text):** A standard BPE tokenizer trained directly on the raw, unsegmented Arabic Wikipedia corpus.

2. **MorphBPE (Pre-segmented Text):** The corpus was first segmented using the Farasa morphological analyzer (Darwish and Mubarak, 2016), and then MorphBPE was trained on this pre-segmented text.

A language model was then trained for each condition on the full corpus. Figure 3 shows the training cross-entropy curves.

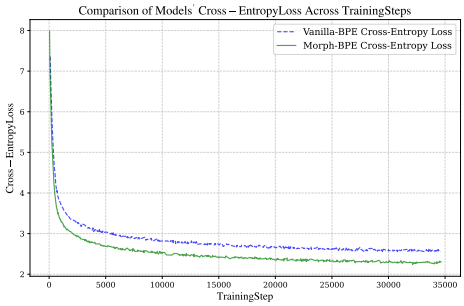

Figure 3: Comparison of Language Model training cross-entropy on the full Arabic Wikipedia corpus. MorphBPE trained on pre-segmented text (red) achieves consistently lower loss than Standard BPE trained on raw text (blue).

The results demonstrate that MorphBPE achieves lower cross-entropy loss than standard BPE. This confirms that the benefits of morphologically coherent segmentation persist when scaling to full-text training scenarios.

B Token-Length Statistics and Fairness

A potential concern when comparing cross-entropy across tokenizers is that models with significantly shorter average token lengths (i.e., higher fertility) might artificially achieve lower per-token cross-entropy. To ensure that the improvements observed with MorphBPE are due to better segmentation quality rather than trivial differences in token granularity, we analyzed token-length distributions derived from large Wikipedia dumps for English, Hungarian, and Russian under the fixed vocabulary sizes used in our experiments.

Table 3 presents descriptive statistics of token lengths. MeanLen is the average token length in characters, and WMeanLen is the frequency-weighted mean token length. The results show that MorphBPE and BPE produce extremely similar token-length characteristics. In all three languages, the difference in mean and weighted mean token lengths between BPE and MorphBPE is miniscule (0.05–0.14 characters). This empirical evidence confirms that the experimental comparison is fair and that cross-entropy gains are attributable to improved morphological alignment rather than token-length artifacts.

Table 3: Wikipedia-derived token-length and distribution statistics comparing BPE and MorphBPE under fixed vocabulary sizes. The similarity in lengths confirms the fairness of cross-entropy comparisons.

Language / Tokenizer	Tokens	Vocab	MeanLen	WMeanLen	μ	σ	α
English / MorphBPE	7B	16K	5.23	2.87	9.79	2.68	1.01
English / BPE	7B	16K	5.28	2.91	9.87	2.64	1.01
Hungarian / MorphBPE	500M	16K	6.14	2.65	6.83	2.80	1.01
Hungarian / BPE	500M	16K	6.20	2.78	7.17	2.71	1.01
Russian / MorphBPE	3B	16K	6.41	2.42	7.64	3.38	1.01
Russian / BPE	3B	16K	6.27	2.64	8.17	3.06	1.01