MYTE: Morphology-Driven Byte Encoding for Better and Fairer Multilingual Language Modeling

Anonymous ACL submission

Abstract

A major consideration in multilingual language modeling is how to best represent languages with diverse vocabularies and scripts. Although contemporary text encoding methods cover most of the world's writing systems, they exhibit bias towards the high-resource languages of the Global West. As a result, texts of underrepresented languages tend to be segmented into long sequences of linguistically meaningless units. To address the disparities, we introduce a new paradigm that encodes the same information with segments of consistent size across diverse languages. Our encoding convention (MYTE) is based on morphemes, as their inventories are more balanced across languages than characters, which are used in previous methods. We show that MYTE produces shorter encodings for *all 99* analyzed languages, with the most notable improvements for non-European languages and non-Latin scripts. This, in turn, improves multilingual LM performance and diminishes the perplexity gap throughout diverse languages. [1]

1 Introduction

Multilingual language models have become the state-of-the-art solution for performing tasks on a wide range of languages (Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021). However, it is challenging to ensuring high performance for all languages, due to differences in data availability especially for the long tail of low-resource languages (Malkin et al., 2022). This challenge is compounded by choices of how words are represented in different langauges during tokenization; past studies showed that multilingual models either cannot accurately represent texts in rare languages (Pfeiffer et al., 2021) or do so via over-segmentation, which is detrimental both to model performance and inference cost (Petrov et al., 2023; Ahia et al., 2023).

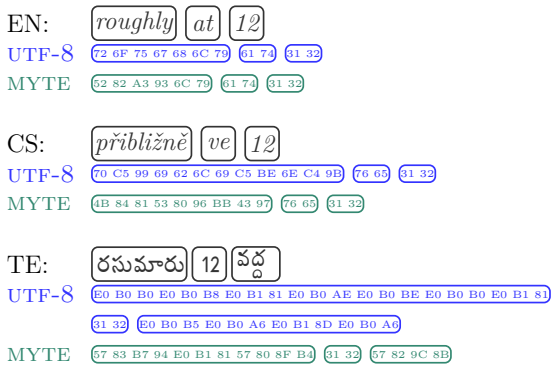


Figure 1: The same phrase is spelled in three languages: English, Czech, and Telugu. *UTF-8* byte encoding of the phrase is shown in blue, while MYTE in green underneath. MYTE achieves higher encoding compression for texts using diacritics or non-Latin script.

Byte-level models aim to solve these challenges. They use byte-level text representations which increase coverage (Xue et al., 2022), for example common encodings such as UTF-8 support most of the world's scripts. Nevertheless, the over-segmentation problem still exists even at the byte level, as byte sequences for single characters are overly long for many low-resource and non-Latin script languages. This problem has an immense effect on modeling these scripts in NLP systems, as operating on longer sequences significantly increases the computation costs of training and inference in models, while also making learning less sample efficient. Furthermore, the billing for APIs such as ChatGPT (`openai.com/chatgpt`) is often associated with the segmented sequence length, disadvantaging speakers of specific languages (Ahia et al., 2023).

In this work, we propose a novel method to derive byte representations of text and enable equitable segmentations across languages and scripts. In our approach, we replace the current conven-

[1] The code and models will be released upon publication of this work.

tion of assigning byte codes to characters with a morphology-driven approach, as morphemes[2] are more informatively comparable constituents of text across languages than characters (Cotterell et al., 2018). Specifically, we introduce a novel algorithm for representing text as byte sequences that is based on unsupervised morphological segmentation (Smit et al., 2014). We demonstrate that our new paradigm for byte representation improves the segmentation of diverse languages of various scripts and morphological inventories. Furthermore, the segmentation of parallel sentences across languages converges to comparable lengths.

We test our method's effectiveness in creating equitable text representation – representations that given parallel texts have similar encoded sequence lengths. We then evaluate the applicability of the method to multilingual language modeling across 99 typologically diverse languages.

Our contributions can be summarized as follows: (a) We propose a novel byte-encoding method that is morphologically driven; (b) We show empirically that the resulting representations are more equitable across languages than vanilla byte, character, or subword segmentation; (c) We analyze the typical lengths of these representations and show decreased sequence length across all analyzed languages, significantly reducing computation cost and benefiting non-Latin script languages the most; (d) We train a language model with our new representation scheme and demonstrate that it maintains balanced and better LM performance across diverse languages and exhibits faster inference speed. This improvements holds across different model scales.

We will release our code and models to facilitate further research in this direction.

2 Background: UTF-8 Bytes

The vast majority of texts online[3] are represented as bytes via *UTF-8* convention, which is defined by the Unicode Standard (The Unicode Consortium, 2011). In *UTF-8*, each character (or codepoint) is represented as a sequence of one to four bytes. Due to the gradual development of communication standards, *UTF-8* first allocated one-byte representation ASCII symbols, which cover primary Latin-script characters (see 00 to 7F in Figure 2). Other characters are represented as multi-byte codes starting with a byte from range C2 to F4 denoting the number of bytes in the codepoint and followed by continuation bytes from range 80 to BF.

In *UTF-8* convention, characters in non-Latin alphabetic scripts (Cyrillic, Armenian, Georgian), diacritics, and abjads[4] usually have two-byte codes, while the byte length increases to three or four for Brahmic abugidas[5] and CJK (Chinese, Japanese, Korean) logographs. As a result, the granularity of byte codes varies significantly across languages; this means that texts conveying the same information across languages tend to be represented by byte sequences of significantly different lengths.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1x	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2x	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6x	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL
8x	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
9x	+10	+11	+12	+13	+14	+15	+16	+17	+18	+19	+1A	+1B	+1C	+1D	+1E	+1F
Ax	+20	+21	+22	+23	+24	+25	+26	+27	+28	+29	+2A	+2B	+2C	+2D	+2E	+2F
Bx	+30	+31	+32	+33	+34	+35	+36	+37	+38	+39	+3A	+3B	+3C	+3D	+3E	+3F
Cx	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Dx	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Ex	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Fx	4	4	4	4	4	4	4	4	5	5	5	5	6	6		

Figure 2: UTF-8 codepage (inspired by the visualizations from: en.wikipedia.org/wiki/UTF-8). Each row contains 16 bytes with the same leading hexadecimal digit. Bytes in the range C2-F4 are leading bytes. They mark the beginning of a multibyte code of the length shown in each cell. Bytes in the range 80-BF are continuation bytes, which follow a leading byte in multibyte codes. Bytes FE and FF are unused. Range C2-F4 encodes Latin capital letters. In MYTE, these characters are decomposed to free space used to encode morphemes.

[2]In this work, the usage of term "morphemes" encompasses both "morphemes" and "morphs". Some linguistic theories use the term "morph" for specific textual realizations of abstract "morphemes". For instance, in English, *es* as in *foxes* and *s* as in *cats* are two distinct "morphs" of a plurality "morpheme". For an in-depth discussion about these two terms, see Section 4 of Žabokrtský et al. (2022)

[3]https://w3techs.com/technologies/overview/character_encoding

[4]Abjads are writing scripts that do not denote vowels, e.g., Hebrew, Arabic.

[5]Abugidas are scripts representing consonant-vowel as one character, typical to the Indian Subcontinent and South-East Asia, e.g., Devanagari, Bengali.

3 Method: Morphology-Driven Bytes

As discussed in the prior section and shown in Figure 1, *UTF-8* convention produces longer byte sequences for some languages due to the development choices. To make byte representation more equitable, we introduce an encoding paradigm that aims to assign byte codes of similar lengths to morphemes across languages. We base our encoding scheme on morphological analysis because morphemes are the shortest meaningful constituents and are independent of the writing convention (Haspelmath and Sims, 2010). We assume that the number of morphemes in sentences with the same information load is more balanced across languages than the number of characters, bytes, or tokens. Thus, we enforce balanced segmentation granularity across languages.

An alternative approach to encoding morphological representations would be treating the union of multilingual morpheme inventories across languages as one large subword vocabulary. To cover the morphemes of many languages in this manner, the vocabulary would be much larger than the ones usually applied to models.[6] This would incur additional computational costs and, similar to other subword representations, would likely not generalize well to new, unseen languages.

3.1 Morphological Analysis

We train an unsupervised morphological analyzer, Morfessor (Smit et al., 2014) on lexicons derived from whole Wikipedia articles in 99 languages. The morphological analysis is performed on each of the languages separately to balance the number of morphemes per language, regardless of data resourcefulness. For each language, we derived a set of 4096 morphemes; the number was chosen to balance segmentation granularity across languages. For each morpheme, we save its score, defined as the hypothetical loss reduction of the Morfessor model if the morpheme had not been included in the set. We take the union of sets across languages to obtain a multilingual morpheme inventory. The details of lexicon preparation and the usage of Morfessor are provided in Appendix A.

[6]The proposed MYTE encoding offers capacity for 2,130,432 of variable length codepoints. It is considerably more than in any of the commonly used subword vocabularies. For reference, large vocabulary XLM-V model allocates 1 million subwords (Liang et al., 2023).

ID	Group	Unicode Script(s)	Leading Byte		
			2 b	3 b	4 b
0	Latin	Latin	42	4A	52
1	Common	Mixed, Common, Inherited, Unkown	43	4B	53
2	Non-Latin Alphabetic	Greek, Cyrillic, Armenian, Georgian	44	4C	54
3	Abjads	Hebrew, Arabic, Syriac, Thaana, Tifinagh	45	4D	55
4	Abugidas North	Devanagari, Gurmukhi, Gujarati, Oriya, Bengali, Sinhala, Tibetan	46	4E	56
5	Abugidas South	Telugu, Kannada, Tamil, Malayalam, Thai, Lao, Myanmar, Tai, Tagalog, Khmer	47	4F	57
6	CJK	Hangul, Han, Yi, Katakana, Hiragana, Bopomofo	48	50	58
7	Other	Remaining scripts	49	51	59

Table 1: Groups of scripts with the initial bytes for their morphological blocks. The groups were selected to balance the number of covered languages with similar writing systems.

3.2 Enriching Byte Representation with Morphology

To alleviate *UTF-8* inefficiencies, we propose a systematic rearrangement of byte codepage. We free 26 bytes (41 to 5A) by decomposing capital letter codes into lowercase letters and capitalization markers. The first byte from this range (41) is repurposed as a capitalization marker. The remaining 25 bytes are freed space used to store morphemes.

Our method takes the sequences of *UTF-8* bytes and transcodes them into shorter sequences using the vocabulary of the same size, i.e. 256, as depicted in Figure 1. We apply the following steps to transcode *UTF-8* sequences to MYTE encodings:

1. We use *UTF-8* as base encoding of text. Then, the byte sequences are transcoded from left to right, merging morpheme sequences and replacing them as dedicated codepoints described in the following points.

2. The morphemes are grouped by scripts as shown in Table 1. Codepoints of multiple

scripts within a single morpheme are assigned to the second cluster (Mixed script).

3. The morphemes are ranked based on their Morfessor score defined in Section 3.1.

4. We assign multibyte codepoint for each of the morphemes analogously to the *UTF-8* convention (see Section 2). Specifically, the first byte denoting the beginning of the morphological codepoint is assigned from the freed range (42-5A) based on the morph's inclusion in one of the script groups. It is followed by continuation bytes from the 64 element range 80-BF, as in *UTF-8* convention. The 64 morphemes with the highest score are saved as two-byte codepoints, following $64^2 = 4096$ as three-byte codepoints; the remaining morphemes are saved as up to $64^3 = 262,144$ four-byte codepoints. The capacity for new codepoints was not exhausted for any script group.

4 Equitable Multilingual Segmentation with MYTE

We first analyze the properties of our proposed morphology-driven encoding. Following the setting of Petrov et al. (2023), we measure whether MYTE produces the segmented sequences of comparable length across languages.

We compute parity across languages using the multi-parallel corpus Flores 200 (Team et al., 2022). Parity is defined as $|t(s_l)|/|t(s_{en})|$, where s_l and s_{en} stand for parallel sentences in language l and in English, respectively. $|t(s)|$ is the length of sequence s with segmentation method t.

We compare the MYTE encoding from Section 3.2 to several baselines of common input representation: (a) Vanilla byte-level encoding via UTF-8; (b) Character-level encoding; (c) Subwords produced by Sentencepiece algorithm (Kudo and Richardson, 2018). In comparison, we focus on the equitability of sequence lengths produced by the methods for diverse languages.

Furthermore, we compare our morphological byte encoding sequence compression rate against the *UTF-8* convention. Compression is essential for an effective text representation as it affects NLP systems' efficiency and usage cost (Ahia et al., 2023). Finally, we check whether our method more effectively compresses languages and scripts unseen in MYTE algorithm described in Section 3.2.

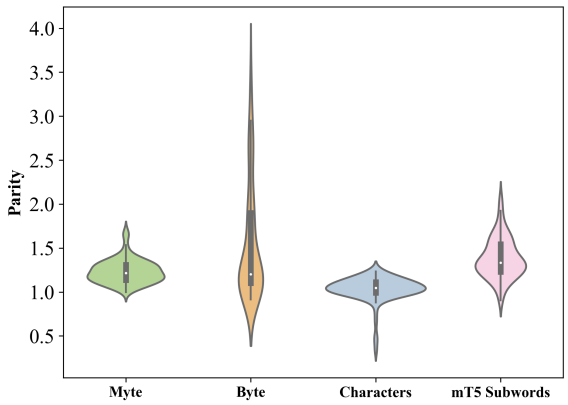


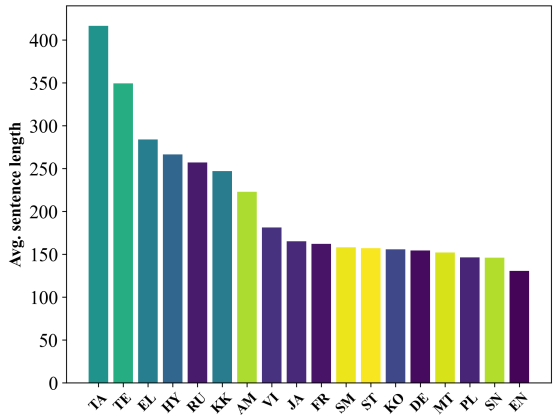
Figure 3: Boxplot aggregating parity against English for three segmentation methods: MYTE, *UTF-8*, characters, and subword tokens from mT5 tokenizer (Xue et al., 2021). Parities were computed on multi-parallel Flores 200 corpus.

4.1 Results

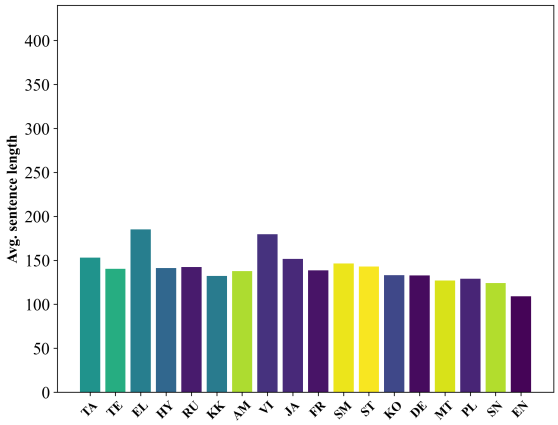
MYTE is Equitable across Languages The comparison of sequence length across parallel sentences in Flores 200 is shown in Figure 4. Our representation is more balanced across languages than the original *UTF-8* bytes. There are still four languages with observably higher code lengths (e.g., Greek, Vietnamese, Punjabi, Khmer). However, MYTE encoding still improves their parity to English such that it is much lower than outlier languages in *UTF-8* (1.7 vs. 3.5 in the worst-case languages, respectively).

Figure 3 shows that MYTE representations are more balanced in parity scores across languages than subword tokenization. In particular, we improve on the long tail of languages over-segmented either in byte or subword encoding. The parties closest to MYTE are obtained by character representation. However, the set of all Unicode characters is larger by orders of magnitude than the number of unique bytes used in MYTE (149,878 vs. 254).

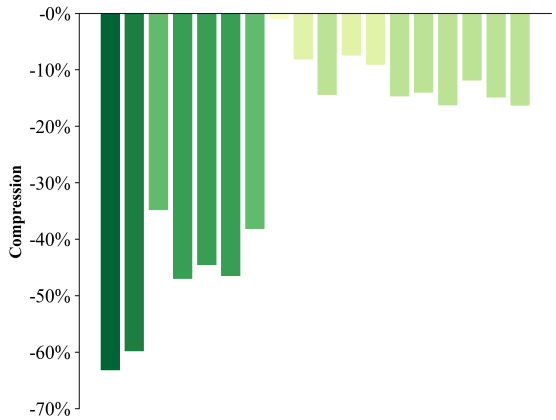
MYTE Encoding Compresses Text Representation The encoded sequence lengths are decreased with MYTE encoding for all languages, as depicted in Figure 4c. The rate of compression varies from 1% for Vietnamese and Chinese to almost 70% for Burmese. As seen in Table 2, the highest compression is obtained for low-resource languages with non-Latin scripts. Notably, this group of languages is the most susceptible to over-segmentation in *UTF-8* encoding.



(a) *UTF-8*



(b) MYTE



(c) Sequence compression

Figure 4: Average byte sequence lengths of parallel sentences from Flores 200 encoded by a) *UTF-8* and b) MYTE. Figure c) depicts the percentage by which the latter sequences are shorter than the former. Results for all the languages can be found in Appendix B.

Findings Generalize to Unseen Languages but not Unseen Scripts In Table 2, we observe that a decrease in sequence length and parity applies to five low-resource languages not considered in constructing MYTE representation, referred to as *unseen languages*. One exemption from the rule is Santhali, written in *unseen* Ol Chiki script, for which we do not observe a change in the encoded sequence length. This observation highlights the importance of considering a wide range of languages and scripts when constructing morpheme inventories. Importantly, MYTE did not reach a capacity of available byte codepoints, and thus, the method can be extended to additional languages. The complete results for *unseen* languages and scripts are shown in Appendix B.

	Byte		Myte		Comp.
	Parity	Len.	Parity	Len.	
English	1.00	131	1.00	109	16%
Latin HR	1.14	149	1.18	129	14%
Latin LR	1.12	147	1.18	128	12%
¬Latin HR	1.62	212	1.29	141	29%
¬Latin LR	2.33	305	1.33	145	50%
Seen	1.56	204	1.24	135	26%
Unseen Lang	1.50	196	1.27	138	23%
Unseen Script	2.80	365	3.35	365	0%
Unseen	1.72	224	1.61	176	19%

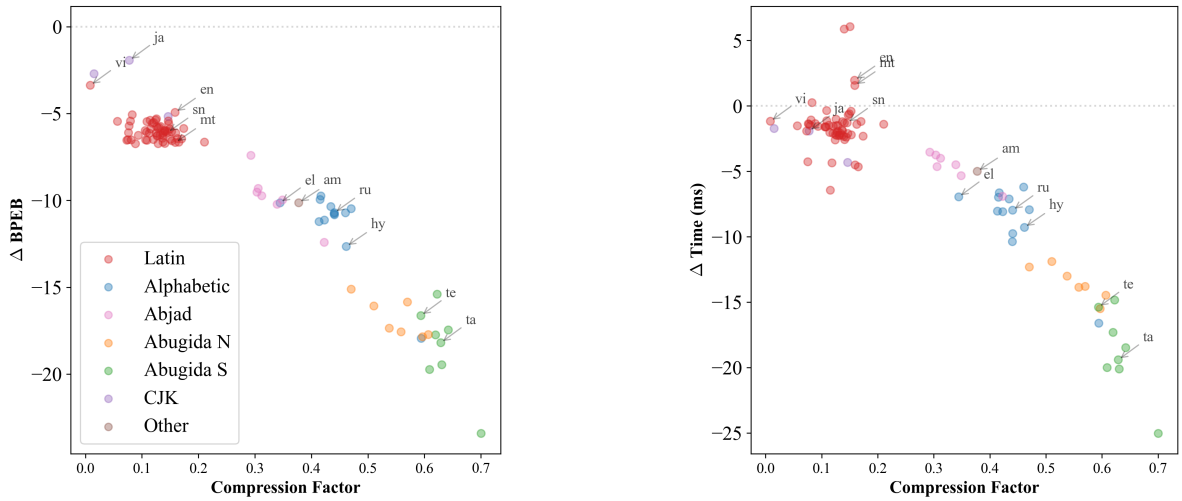
Table 2: Averaged sequence length and corresponding parities to English of *UTF-8* and MYTE. We aggregated results for languages used in morphological adaptation (i.e., *Seen*) by their script (Latin vs. Non-Latin) and resourcefulness (HR: high resource, LR: low resource) based on categorization from Joshi et al. (2020). The last three rows present results for languages *unseen* in morphological adaptation; all of them are low-resource. Shortened column headers: Len. – Length, Comp. – Compression.

5 MyT5: Language Modeling with MYTE

This section investigates the benefits of using MYTE in byte-level language modeling. For that purpose, we have trained T5 language models on MYTE representation. We refer to these models as **My**te **T5** models, or MyT5 for short.

5.1 Training Details

We base the architecture and implementation of our MyT5 model on the byte-level T5 model, i.e., ByT5 (Xue et al., 2022). ByT5, like other T5 models (Raffel et al., 2020), is an encoder-decoder Transformer model trained on predicting masked spans of texts. The main novelty of the ByT5 model is that it op-



(a) LM Performance ($\rho_S = -0.81$)

(b) Inference Time ($\rho_S = -0.77$)

Figure 5: The difference in Byte-per-English-Bit and inference time between MyT5 and ByT5 large models against compression factor of MYTE. For each sentence, the BPEB value is normalized by the number of UTF-8 bytes used to represent the corresponding English sentence. The inference was run on A40 GPU core, we report an average per-example delta. ρ_S are Spearman's correlation coefficients.

erates on bytes instead of subword tokens, as in the standard T5 model, making it a suitable base model for our setting.

We pre-train three new instances of MYTE-level models of different sizes: small (300M), base (582M), and large (1.23B parameters). For pre-training, we used the standard task of restoring corrupted spans from mC4 corpus (Raffel et al., 2020). All the byte sequences are transcoded into morphologically-driven bytes. We use Jax implementation, i.e., t5x repository (Roberts et al., 2022), and the same hyperparameters as in ByT5 (Xue et al., 2022). The only difference from their training approach is that we pre-train for 250,000 steps rather than one million steps since we observe overfitting when training for more steps, especially on low-resource languages. Similarly, Chung et al. (2023) also observed overfitting in multilingual T5 models caused by extensive duplications in the mC4 corpus, leading them to train models for 250,000 steps only. In evaluations, we compare against a reimplemented ByT5 instance trained for the same time.

5.2 Experiments

We compare the performance of the MyT5 and ByT5 models, focusing on two aspects:

First, the multilingual language modeling performance of MyT5 – how is it, and is it comparable across languages? Following Cotterell et al. (2018), we use the Bit-per-English-Byte metric on the multi-parallel FLORES 200 corpus to control for the informativeness of evaluation sequences:

$$BPEB = \frac{1}{|\mathbf{c}_{English,UTF}| + 1} \sum_{i=1}^{|\mathbf{c}|+1} \log p(c_i|\mathbf{c}_{<i}) \quad (1)$$

$\mathbf{c}$ is a sequence of bytes (original *UTF-8* or MYTE) with c_i being the i-th byte. For normalization, we use the number of *UTF-8* bytes in English sentence $\mathbf{c}_{English,UTF}$ for fair comparison across languages and representation methods. It is the main difference from perplexity, which is normalized by the sequence length and thus confounded by segmentation rates characteristic of individual languages and encodings.

Second, we compare inference times of text generation of MyT5 and ByT5. We expect a decrease in sequence length, as shown in the last section, will render up to a quadratic reduction of forward-pass time due to the quadratic complexity of attention computation.

For both aspects, we report the results on three scales of the model (small, base, and large). Unless stated otherwise, we present the results of the large model.

5.3 Results

MyT5 Outperforms ByT5 in Language Modeling In Figure 5a, our model outperforms ByT5, producing lower (better) average $BPEB$ scores for all analyzed languages. The improvement is strongly negatively correlated with the compression rate discussed in the previous section. The gains are largest for languages using Abugidas (scripts representing consonant-vowel as one character, typical to the Indian Subcontinent and SE Asia) that tend to be shortened the most by MYTE encoding. On the other end of compression distribution, we still observe (smaller) improvement for Latin and CJK scripts. This observation suggests that the MYTE encoding's leverage is not constrained to shortening sequences, but it also uses codepoints that are easier to predict by a language model. MYTE uses codepoints based on morphemes that are inherently meaningful language units in contrast to orthographic symbols, which are the backbone of the *UTF-8* convention.

Encoding in MyT5 Diminishes LM Performance Gap Across Languages Previous works observed that some languages are more challenging language models (Cotterell et al., 2018) due to their morphological properties. In contrast, others showed that LM performance is linked with how texts in specific languages are represented (Park et al., 2021). Our results in Figure 6 support the latter view, specifically that the predictability of the languages can be balanced by using equitable underlying representation, i.e., MYTE encoding. For the experiments across languages, we show that MyT5 achieves more balanced $BPEB$ across languages than ByT5. As discussed in the previous section, the improvement is the starkest for languages prone to over-segmentation under *UTF-8*. The worst results of MyT5 were obtained for languages benefited by MYTE to a lesser extent, as observed in Section 4.1: Greek and Vietnamese.

MyT5 is More Efficient across Scales than ByT5 As shown in Figure 5b, MyT5's inference time is shorter than that of ByT5 for almost all languages. This behavior is mostly observed for Non-Latin script languages and can thus be attributed to sequence shortening.

Table 3 shows that MyT5 edge over ByT5 is the highest with the largest available scale. It hints that MYTE can be successively applied to models of larger scales.

		Byt5		Myt5	
		BPEB	T (ms)	BPEB	T (ms)
small	All	10.1	7.0	4.6	6.7
	Latin	4.6	5.9	4.2	6.6
	Non Latin	18.1	8.5	5.1	6.8
base	All	8.2	11.5	5.8	8.9
	Latin	4.9	9.4	5.0	8.7
	Non Latin	13.0	14.6	6.9	9.1
large	All	13.4	31.8	4.6	26.7
	Latin	10.1	28.1	4.0	26.6
	Non Latin	18.2	37.3	5.4	27.0

Table 3: Byte-per-English-Bits and Inference times (average per sentence) averaged for three language groupings.

6 Related Work

6.1 Fair Representation across Languages

Perhaps the most significant challenge of multilingual NLP is the large disparity of resourcefulness across the world's languages (Joshi et al., 2020), as the size and quality of data used for the model training directly affects its performance in individual languages. Hence, researchers have proposed multiple ways to balance the training signal across languages (Malkin et al., 2022). Solutions include sampling data to overrepresent low-resource languages, e.g., with alpha (Conneau et al., 2020) or uniform sampling of data across languages (Chung et al., 2023). This unequal treatment of languages is also present in how data is encoded as input to the model (Ahia et al., 2023). Petrov et al. (2023) show that practically all methods used to represent texts as input of NLP systems treat languages unequally, segmenting some (mainly the lowest-resourced ones) into fine-grained non-informative units.

Some approaches aimed at balancing the segmentation or tokenization methods have been proposed. Limisiewicz et al. (2023) proposed merging vocabulary based on the tokenizer scoring function. Zheng et al. (2021) introduced a method of allocating vocabulary capacity uniformly across languages, while Chung et al. (2020) constructed multilingual vocabulary for clusters of languages and merged them. Liang et al. (2023) combined the elements of both approaches and showed the advantage of extending vocabulary to benefit multilingual transfer. These solutions offer the promise of obtaining a better allocation of vocabulary units. However, they do not solve the inequality of the

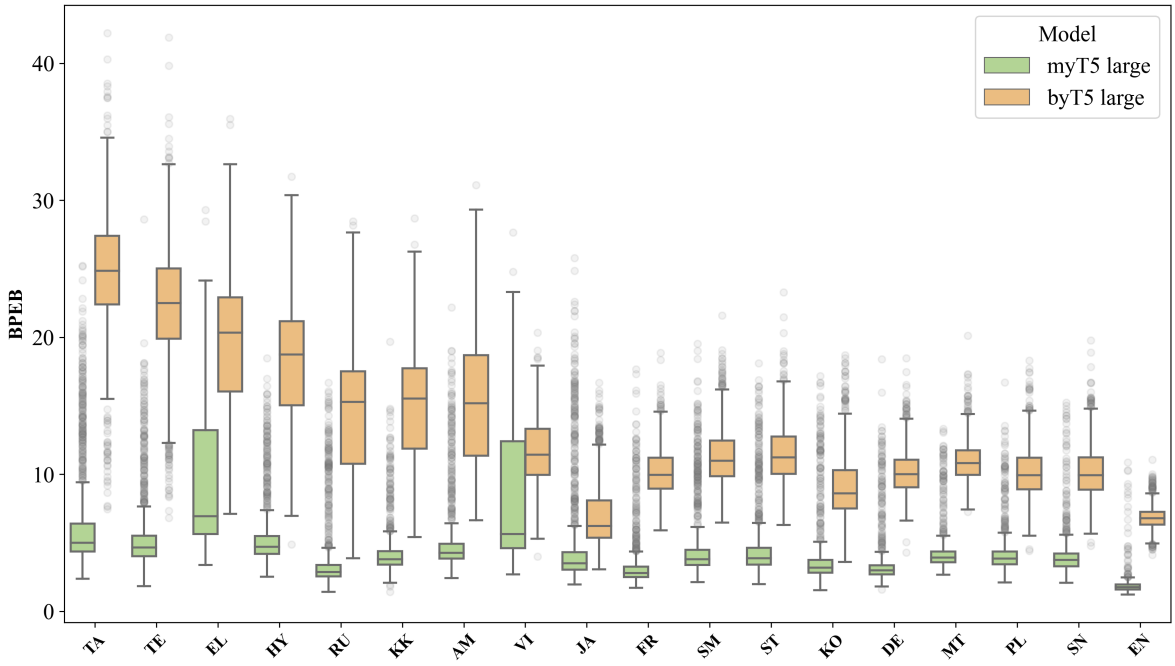


Figure 6: Sentence prediction superisal expressed as the normed NLL score on multi-parallel FLORES-200 corpus. The comparison shows that under myT5 model, surprisal factors are more equitable than in the standard byT5 model.

underlying encoding, which may affect the construction process of vocabulary units. For instance, byte merges in the BPE algorithm always begin at individual bytes (Sennrich et al., 2016; Zouhar et al., 2023).

Morphological analyzers, such as Morfessor, showed promising results for segmenting input texts for language models and neural machine translators (Machácek et al., 2018; Hou et al., 2023). We are the first to apply morphology-based encoding for a massively multilingual setting.

6.2 Tokenization-free Language Modeling

An alternative to subword tokenization is representing texts directly as underlying encoding: characters or bytes. Or even representing texts as pixels of rendered text images (Rust et al., 2023).

Xue et al. (2022) shows that for many non-Latin scripts, byte-level encoding performs worse than subword tokenization. The problem with small units is that they do not carry meaningful information independently and often underperform subword models (Sun et al., 2023; Clark et al., 2022).

The researchers have proposed multiple algorithms to enrich the byte-level embeddings with information from a local context. For that purpose, recent approaches use shallow networks to aggregate information in local contexts defined as character n-grams (Clark et al., 2022), byte patches (Yu et al., 2023), or character blocks (Tay et al., 2022). However, the problem with choosing the appropriate context window is hard, because information density varies for different languages. A solution to that problem can be dynamically learning the segmentation in byte sequences (Nawrot et al., 2023). Another approach is to redefine the encoding convention to equate the information loads in sequences, as the proposed MYTE approach.

7 Conclusions

In this paper, we introduce MYTE encoding, a fairer byte-level representation for multilingual language modeling that is based on morphological segmentation. We show that adapting a morphological analyzer to unsupervised segmentation allows us to represent multi-parallel corpora in comparable encoding lengths across a wide range of languages Additionally, our new representation significantly improves modeling, especially of low-resource and non-Latin script languages. This trend holds across models of different sizes.

Ethical Statement

Our work makes a significant contribution to a fairer representation of text across diverse languages. It will potentially benefit the speakers of underrepresented languages by enabling access to more reliable and cheaper NLP tools. For all the experiments, we relied on open-source tools and datasets. We strongly discourage unintended usage of the released language models.

Limitations

Our method inherits the limitations of Morfessor, which was used to obtain multilingual morphological segmentation for MYTE. First, Morfessor is data dependent and is affected by the quality of the corpus (Wikipedia) and the lexicon (MUSE when available). The artifact of these resources is a significant presence of cross-lingual contamination, typically from high-resource languages (Blevins and Zettlemoyer, 2022). This leads to the appearance of Latin (typically English) morphemes in analyses of many languages. Second, we use the unsupervised mode of Morfessor that can be applied to any language due to its independence of annotated data. However, it is also prone to errors in morphological segmentations, i.e., oversegmenting texts of specific languages. We mitigate this issue by picking a constant target number of morphemes.

Dependence on data might also affect the generalizability of our findings' to the languages that were not used in the construction of MYTE. Results in Section 4.1 show that the method is indeed effective in compressing text representation of unseen languages but not unseen scripts. Notably, we do not exhaust the capacity of the MYTE codepage; thus, it can be extended to further languages.

Lastly, even perfect morphological analysis cannot guarantee equal granularity of segmentation across languages. Some languages are characterized by higher morphological richness, thus their texts consist of more morphemes. Accordingly, we observe differences in MYTE segmentation lengths across languages, yet these disparities are significantly smaller than in other conventions.

References

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David R. Mortensen, Noah A. Smith, and Yulia Tsvetkov. 2023. Do All Languages Cost the Same? Tokenization in the Era of Commercial Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 9904–9923. Association for Computational Linguistics.

Terra Blevins and Luke Zettlemoyer. 2022. Language Contamination Helps Explains the Cross-lingual Capabilities of English Pretrained Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hyung Won Chung, Xavier Garcia, Adam Roberts, Yi Tay, Orhan Firat, Sharan Narang, and Noah Constant. 2023. UniMax: Fairer and More Effective Language Sampling for Large-scale Multilingual Pretraining. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving Multilingual Models with Language-clustered Vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4536–4546. Association for Computational Linguistics.

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Ryan Cotterell, S. J. Mielke, Jason Eisner, and Brian Roark. 2018. Are All Languages Equally Hard to Language-Model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 536–541. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

M. Haspelmath and A.D. Sims. 2010. *Understanding Morphology*. Understanding language series. Hodder Education.

Jue Hou, Anisia Katinskaia, Anh-Duc Vu, and Roman Yangarber. 2023. Effects of sub-word segmentation on performance of transformer language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7413–7425. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

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

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13142–13152. Association for Computational Linguistics.

Tomasz Limisiewicz, Jirí Balhar, and David Marecek. 2023. Tokenization Impacts Multilingual Language Modeling: Assessing Vocabulary Allocation and Overlap Across Languages. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5661–5681. Association for Computational Linguistics.

Dominik Machácek, Jonás Vidra, and Ondrej Bojar. 2018. Morphological and Language-agnostic Word Segmentation for NMT. In *Text, Speech, and Dialogue - 21st International Conference, TSD 2018, Brno, Czech Republic, September 11-14, 2018, Proceedings*, volume 11107 of *Lecture Notes in Computer Science*, pages 277–284. Springer.

Dan Malkin, Tomasz Limisiewicz, and Gabriel Stanovsky. 2022. A balanced data approach for evaluating cross-lingual transfer: Mapping the linguistic blood bank. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4903–4915, Seattle, United States. Association for Computational Linguistics.

Piotr Nawrot, Jan Chorowski, Adrian Lancucki, and Edoardo Maria Ponti. 2023. Efficient Transformers with Dynamic Token Pooling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6403–6417. 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. *Trans. Assoc. Comput. Linguistics*, 9:261–276.

Aleksandar Petrov, Emanuele La Malfa, Philip H. S. Torr, and Adel Bibi. 2023. Language Model Tokenizers Introduce Unfairness Between Languages. *CoRR*, abs/2305.15425.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-text Transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. 2022. Scaling Up Models and Data with $\texttt{t5x}$ and $\texttt{seqio}$. ArXiv:2203.17189 [cs].

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language Modelling with Pixels. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Jimin Sun, Patrick Fernandes, Xinyi Wang, and Graham Neubig. 2023. A Multi-dimensional Evaluation of Tokenizer-free Multilingual Pretrained Models. In *Findings of the Association for Computational Linguistics: EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pages 1680–1690. Association for Computational Linguistics.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Prakash Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2022. Charformer: Fast Character Transformers via Gradient-based Subword Tokenization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No Language Left Behind: Scaling Human-centered Machine Translation.

The Unicode Consortium. 2011. The Unicode Standard. Technical Report Version 6.0.0, Unicode Consortium, Mountain View, CA. ISBN: 978-1-936213-01-6.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a Token-free Future with Pre-trained Byte-to-byte Models. *Trans. Assoc. Comput. Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Lili Yu, Daniel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. MEGABYTE: Predicting Million-byte Sequences with Multiscale Transformers. *CoRR*, abs/2305.07185.

Zdeněk Žabokrtský, Niyati Bafna, Jan Bodnár, Lukáš Kyjánek, Emil Svoboda, Magda Ševčíková, and Jonáš Vidra. 2022. Towards universal segmentations: UniSegments 1.0. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1137–1149, Marseille, France. European Language Resources Association.

Bo Zheng, Li Dong, Shaohan Huang, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song, and Furu Wei. 2021. Allocating Large Vocabulary Capacity for Cross-lingual Language Model Pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3203–3215. Association for Computational Linguistics.

Vilém Zouhar, Clara Meister, Juan Luis Gastaldi, Li Du, Tim Vieira, Mrinmaya Sachan, and Ryan Cotterell. 2023. A Formal Perspective on Byte-pair Encoding. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 598–614. Association for Computational Linguistics.

A Details of Unsupervised Morphological Analysis

In this appendix, we provide details on the prerequisites of MYTE transcoding algorithm: a) preparing multilingual lexicons and corpora for morphological analysis and b) usage of Morfessor unsupervised algorithm to obtain morpheme inventory for each language.

A.1 Preparing Lexicons for Morphological Analysis

To obtain morphological segmentation across a wide variety of languages and scripts, we perform the following steps:

1. We use 45 languages with bilingual lexicons available through MUSE (Lample et al., 2018) as a base. Lexicons are obtained independently for each language; hence, we ignore the bilingual aspect of the data. We filter out the lexemes that are the same in English and the target language to avoid contamination that would unfairly boost the frequency of English words across lexicons.

2. We use Wikipedia corpus dump from September 2023 `dumps.wikimedia.org` to count the occurances of lexemes. For 54 languages included in mC4 (Raffel et al., 2020), but without the MUSE lexicon, we compile the list of unique words in Wikipedia as a lexicon.

3. The lexicons are clipped to the size of 30,000 lexemes.

4. All lexemes are transcribed to bytes via *UTF-8* standard. All byte sequences are decomposed following NKFD convention, i.e., modifying symbols (diacritics, accents), which are represented as separate codepoints. On top of *UTF-8* decomposition, we rewrite capital letter codes into lowercase letters and capitalization markers.

A.2 Unsupervised Segmentation with Morfessor

We use Morfessor (Smit et al., 2014), an unsupervised algorithm producing segmentation on a subword level that resembles morphological analysis. The unsupervised nature of the method allows us to apply it to a wide range of languages. However, it is essential to note that the method is prone to errors, such as over-segmentation of roots or misplaced morph boundaries. We use adaptive loss weighting to limit the number of attested morphs to around 4096 to avoid over-segmentation. Unlike the typical usage of Morfessor, we applied it to the corpus on byte instead of character level.

A.3 Morfessor: Technical Details

Morfessor uses recursive optimization to produce subword segmentation akin to morphological analysis. The input data required for unsupervised analysis are language corpus and lexicon consisting of unique words $c \in \mathcal{C}$. We also define the set of atoms $a \in \mathcal{A}$, which are indivisible segments of texts that can be assembled into words. We choose atoms to be *UTF-8* bytes.

The aim of the algorithm is to find a set of morphemes $m \in \mathcal{M}$ appearing in the segmentation of words from the given lexicon. The set of morphemes $\mathcal{M}$ is extended by a recursive algorithm optimizing two losses: *corpus loss* and *lexicon loss* computed with respective data resources. Before providing equations for the mentioned losses, let's define the auxiliary variables:

$$\begin{aligned} M &= \sum_{m \in \mathcal{M}} \#_{\text{COR}}(m) \\ C &= \sum_{c \in \mathcal{C}} \#_{\text{COR}}(m) - 1 \\ A &= \sum_{a \in \mathcal{A}} \#_{\mathcal{M}}(a) \end{aligned} \tag{2}$$

The $\#$ notation is used to denote the number of elements in the corpus (COR) or morpheme set $\mathcal{M}$. In other words, M is the total number of morphemes in the corpus, C is the total number of words in the corpus, and A is the total number of atoms in the set of (unique) morphemes. Morfessor uses the following losses in recursive optimization:

Corpus loss favors morphemes frequently appearing in the corpus:

$$\begin{aligned} \mathcal{L}_{\text{COR}} = (M + C)\log(M + C) + \\ - \sum_{m \in \mathcal{M}} \#_{\text{COR}}(m) \log \#_{\text{COR}}(m) + \\ + \log \binom{M - 1}{|\mathcal{M}| - 1} \end{aligned} \tag{3}$$

Lexocon loss favors segments consisting of diverse sets of atoms so that overlapping segments are not identified as morphemes:

$$\mathcal{L}_{\text{LEX}} = (A + |\mathcal{M}|)\log(A + |\mathcal{M}|) - |\mathcal{M}|\log|\mathcal{M}| + \\ - \sum_{a \in \mathcal{A}} \#_{\mathcal{M}}(a)\log\#_{\mathcal{M}}(a) - log(|\mathcal{M}|!) + \\ + \log\binom{A-1}{|\mathcal{A}|-1} \tag{4}$$

The losses are weighted by a parameter α, which indirectly controls the size of the morpheme set $|\mathcal{M}|$. For instance, we adapt α to keep the number of morphemes close to 4096 for each language. We observed that this size leads to comparable segmentation across languages,

$$\mathcal{L} = \alpha\mathcal{L}_{\text{COR}} + \mathcal{L}_{\text{LEX}} \tag{5}$$

B Supplementary Results

This appendix summarizes complementary results referred to throughout the papers.

Results for Each Language All the experimental results for each of the analyzed mC4 languages are presented in Table 4. Sequence lengths under *UTF-8* and MYTE are visualized in Figure 8. Corresponding compression rates in Figure 9.

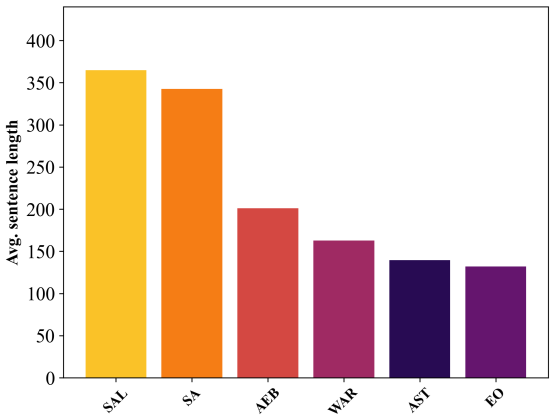
Figure 7 illustrates the sequence lengths and compressions obtained for languages *unseen* in the morphological analysis.

LM Performance Across Scales Figure 10 shows the difference of $BPEB$ between MyT5 and ByT5 the models in small and base scales. Furthermore, Table 4 contains average language modeling scores and inference times across all available scales for each language.

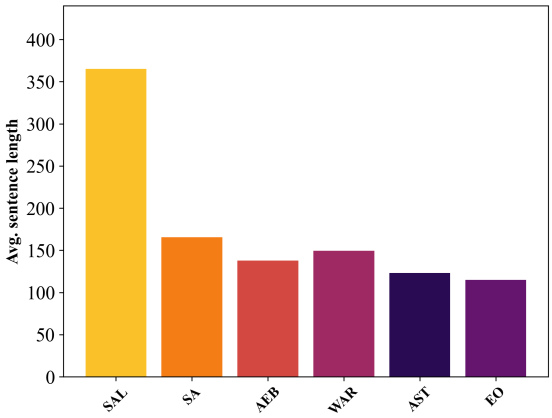
C Compuatational Infrastracture

The MyT5 and reimplemented ByT5 models were trained on TPUs available at the Google Cloud Platform. We used v3-8 for training small and base models and v3-32 for the large model. The training took approximately 90h for small, 230h for base, and 190h for large models. We are thankful to Google for providing free quotas for those machines through the TPU Research Cloud program.

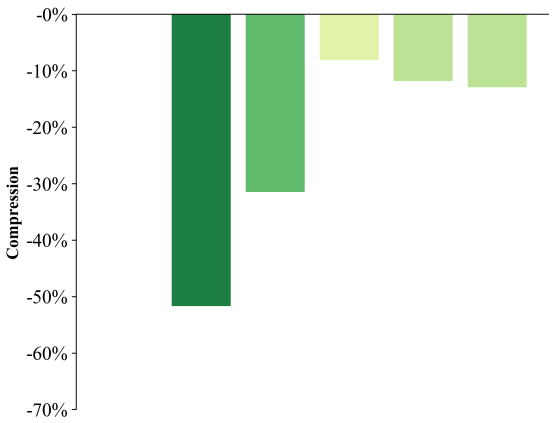
The inference in language modeling experiments was run on an A40 GPU core.



(a) *UTF-8*

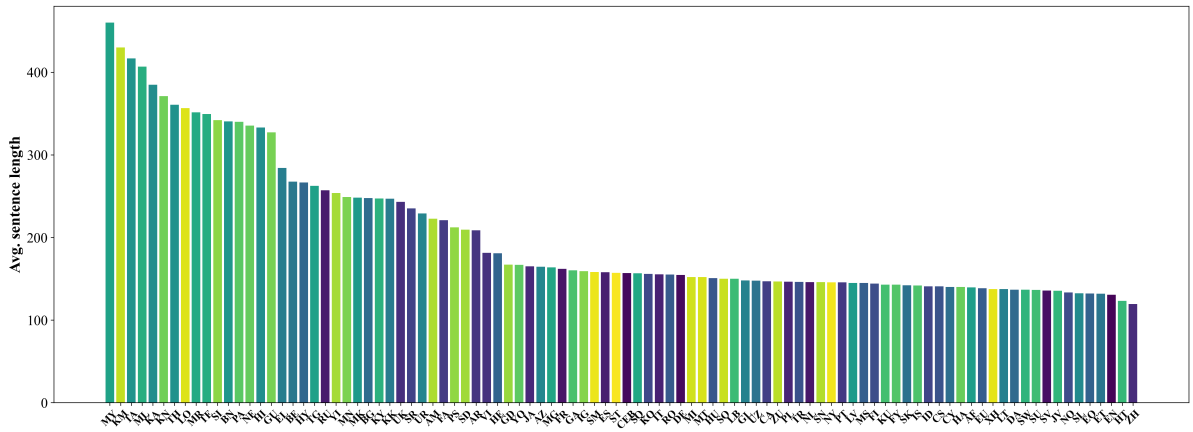


(b) MYTE

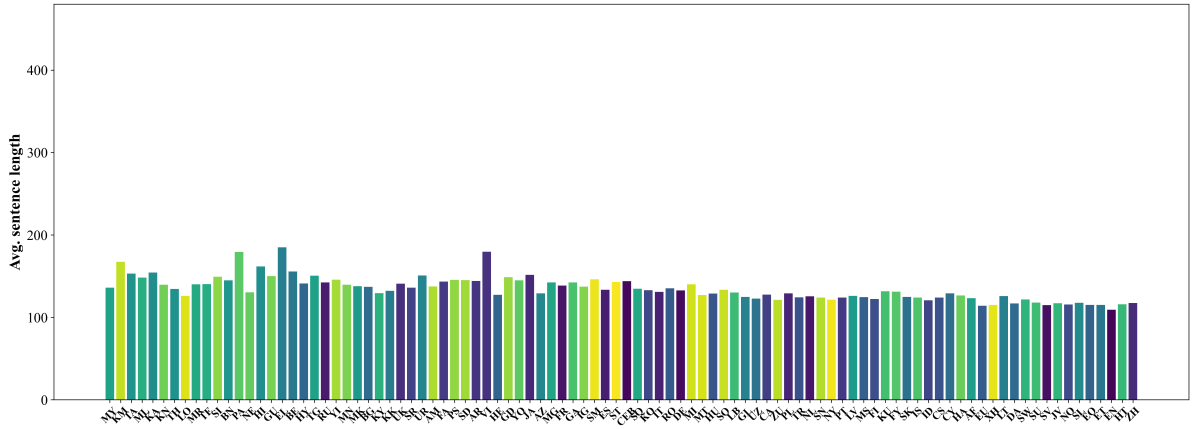


(c) Sequence compression

Figure 7: Average byte sequence lengths of parallel sentences for languages unseen in the morphological analysis. Santali (*sal*) uses a script (Ol Chicki) distinct from all other languages included in morphological analysis.



(a) Original UTF-8



(b) With Morph Codepoints (Same order as in a)

Figure 8: Average byte sequence lenghts of parallel sentences from FLORES-200 (Team et al., 2022)

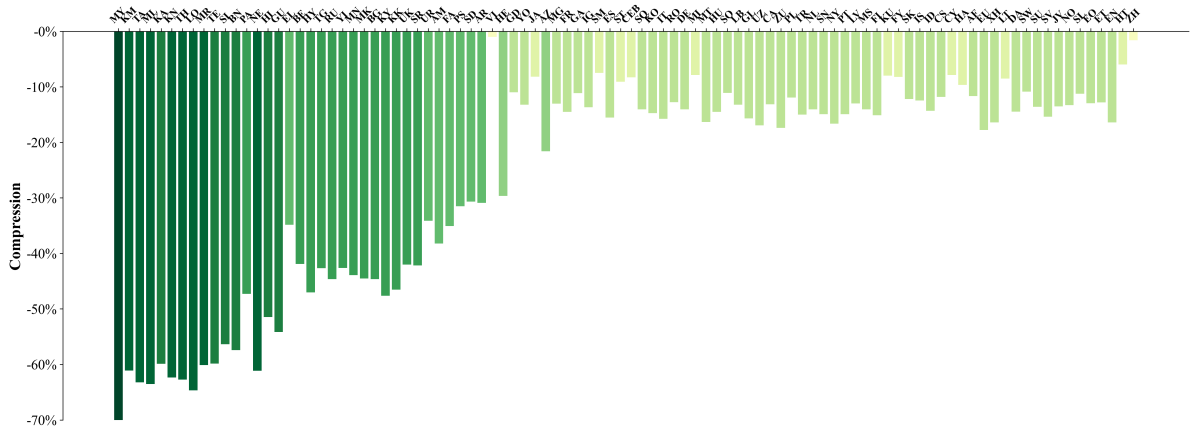
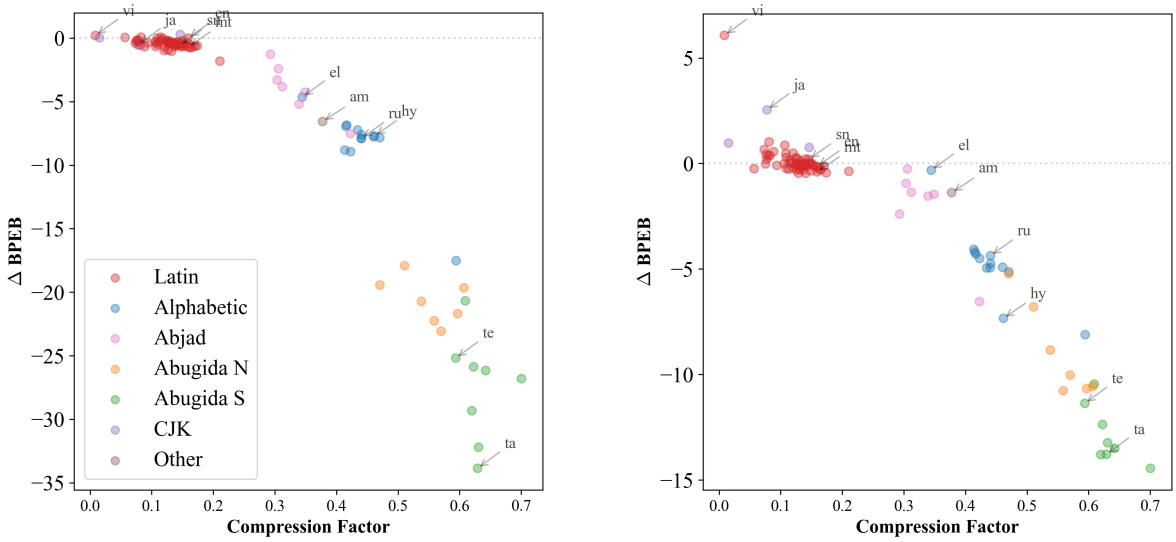


Figure 9: Sequence compression rates on FLORES 200 from decomposition and subsequent morph merging in comparison with the original utf-8 encoding. Ordered as in a



(a) small

(b) base

Figure 10: The difference in Byte-per-English-Bit between MyT5 and ByT5 for models in small and base scales.

Lang	*UTF-8*		MYTE		Comp.	ByT5						ByT5					
	Parity	Len.	Parity	Len.		BPEB			Time (ms)			BPEB			Time (ms)		
					in %	small	base	large	small	base	large	small	base	large	small	base	large
af	1.1	139.6	1.1	123.3	11.7	3.9	4.6	9.5	5.8	8.9	27.6	3.7	4.3	3.4	6.6	8.4	26.1
am	1.7	222.8	1.3	137.6	38.2	11.7	8.3	15.3	6.9	12.5	32.0	5.1	6.9	5.2	6.7	9.0	27.0
ar	1.6	208.8	1.3	144.2	30.9	7.0	6.7	13.7	7.0	11.6	31.4	4.6	6.4	4.4	6.4	9.1	26.7
az	1.3	164.6	1.2	129.1	21.6	6.4	5.7	11.4	6.3	9.9	28.9	4.6	5.4	4.7	6.7	8.7	27.5
be	2.1	267.7	1.4	155.5	41.9	14.6	11.6	17.1	8.1	13.5	35.5	5.7	7.5	5.9	6.9	9.5	27.4
bg	1.9	247.6	1.3	137.1	44.6	11.9	9.6	14.6	7.7	12.6	33.9	4.3	4.9	3.8	7.1	8.8	24.1
bn	2.6	340.6	1.3	145.0	57.4	28.4	17.2	21.3	9.3	16.6	41.3	5.3	7.2	5.4	6.9	9.2	27.5
ca	1.1	147.1	1.2	127.7	13.2	4.1	4.6	9.6	5.8	9.4	27.9	3.9	4.6	3.4	6.7	8.5	25.9
ceb	1.2	156.9	1.3	143.8	8.3	5.0	5.0	11.1	6.0	9.6	28.7	4.4	6.0	4.5	6.3	9.2	27.6
cs	1.1	140.7	1.1	124.1	11.8	4.4	4.8	9.5	5.5	9.0	28.1	4.5	4.8	4.1	6.7	8.4	26.0
cy	1.1	140.1	1.2	129.1	7.9	4.5	5.1	10.7	5.9	8.7	27.6	4.2	5.1	4.2	7.1	8.5	23.4
da	1.0	136.7	1.1	116.9	14.5	3.9	4.5	9.3	5.9	8.7	27.8	3.6	4.0	3.3	6.8	8.2	33.7
de	1.2	154.5	1.2	132.8	14.1	4.6	5.1	10.2	6.2	9.7	27.9	4.2	5.1	3.6	6.5	8.8	26.7
el	2.2	284.1	1.7	185.0	34.9	12.8	13.6	19.5	8.3	14.0	36.6	8.2	13.3	9.3	7.1	10.4	29.7
en	1.0	130.5	1.0	109.1	16.4	2.6	3.3	6.8	6.2	11.0	28.4	2.6	3.2	1.9	6.6	10.5	30.3
eo	1.0	132.2	1.1	115.1	12.9	3.8	4.4	9.1	5.8	8.5	27.2	3.6	4.1	3.3	6.6	8.1	25.7
es	1.2	158.0	1.2	133.5	15.5	4.7	4.8	9.9	6.2	9.2	28.2	4.0	4.8	3.4	6.4	8.7	34.2
et	1.0	131.9	1.1	115.0	12.8	4.1	4.4	9.1	5.8	8.6	27.3	3.9	4.3	3.7	6.4	8.2	24.7
eu	1.1	138.6	1.0	114.0	17.8	4.3	4.6	9.4	5.9	8.8	27.9	3.7	4.1	3.5	6.5	8.1	25.6
fa	1.7	220.9	1.3	143.4	35.1	8.6	7.6	14.5	8.7	12.0	32.0	4.4	6.1	4.5	7.0	9.0	26.7
fi	1.1	144.1	1.1	122.3	15.1	4.7	4.8	9.7	5.8	9.4	26.6	4.2	4.9	4.1	6.7	8.4	25.9
fr	1.2	162.1	1.3	138.6	14.5	4.7	5.1	10.2	6.3	9.6	28.2	4.2	5.3	3.5	6.5	9.0	26.9
fy	1.1	143.0	1.2	131.3	8.2	4.8	5.1	9.8	5.7	9.1	27.6	4.5	5.5	4.4	6.5	8.8	26.1
ga	1.2	160.3	1.3	142.5	11.1	5.4	5.7	11.2	6.0	9.7	28.1	5.0	6.2	5.1	6.7	9.1	26.5
gd	1.3	167.2	1.4	148.8	11.0	5.8	6.0	12.0	6.1	10.5	28.9	5.1	6.9	5.5	6.8	9.1	27.3
gl	1.1	148.0	1.1	124.8	15.7	4.2	4.5	9.4	6.1	9.4	28.0	3.8	4.4	3.3	6.7	8.5	27.6
gu	2.5	327.1	1.4	150.0	54.1	26.5	16.6	23.5	8.9	16.7	39.8	5.8	7.8	6.2	6.8	9.4	26.8
ha	1.1	140.0	1.2	126.5	9.6	4.1	4.6	9.9	5.7	9.3	27.5	3.7	4.5	3.7	6.5	8.6	25.9
he	1.4	180.9	1.2	127.3	29.6	5.6	7.4	11.3	6.5	10.2	29.6	4.3	5.0	3.9	6.7	8.5	26.0
hi	2.6	333.1	1.5	161.6	51.5	23.8	15.8	22.2	9.2	16.2	40.4	5.9	9.0	6.1	7.2	9.7	28.5
ht	0.9	123.2	1.1	115.8	6.0	3.5	4.2	8.7	5.4	8.4	27.1	3.5	4.0	3.3	6.3	8.0	25.6
hu	1.2	150.8	1.2	128.9	14.5	5.2	5.3	10.5	5.8	9.5	28.4	4.7	5.3	4.4	6.7	8.6	25.9
hy	2.0	266.5	1.3	141.2	47.0	13.0	13.6	18.1	8.1	13.4	35.6	5.3	6.2	5.5	7.0	9.0	26.3
id	1.1	140.8	1.1	120.7	14.3	4.0	4.3	9.4	5.9	9.2	27.6	3.5	4.1	3.4	6.6	8.3	25.3
ig	1.2	159.1	1.3	137.4	13.7	5.8	5.7	11.1	6.0	10.5	28.1	4.8	5.8	4.8	6.7	9.0	25.9
is	1.1	141.8	1.1	124.1	12.4	4.9	5.1	9.7	5.7	9.1	27.6	4.2	4.9	4.1	6.4	8.6	26.0
it	1.2	155.4	1.2	130.9	15.8	4.5	4.9	10.0	6.1	9.3	28.1	3.9	4.7	3.4	6.4	8.5	25.9
ja	1.3	165.1	1.4	151.6	8.2	5.4	5.0	7.0	6.3	10.0	28.4	4.9	7.5	5.1	6.7	9.2	26.5
jv	1.0	135.6	1.1	117.2	13.5	3.9	4.3	9.2	5.6	8.9	27.6	3.5	4.0	3.4	6.4	7.9	25.3
ka	2.9	385.0	1.4	154.4	59.9	23.5	16.0	24.5	10.3	18.9	44.9	6.0	7.9	6.6	7.1	9.5	28.3
kk	1.9	247.0	1.2	132.1	46.5	12.2	9.7	15.0	7.7	12.6	33.8	4.4	4.8	4.3	6.3	8.7	27.6
km	3.3	430.0	1.5	167.3	61.1	27.6	22.7	29.0	10.8	20.6	48.3	7.0	12.3	9.3	7.2	10.1	28.3
kn	2.8	371.0	1.3	139.7	62.4	34.1	20.5	23.5	9.8	18.4	43.7	4.8	6.7	5.7	6.7	9.1	26.4
ko	1.2	155.9	1.2	133.0	14.7	4.5	5.0	9.1	6.0	9.7	30.6	4.7	5.8	4.0	6.8	8.7	26.2
ku	1.1	143.0	1.2	131.6	8.0	4.7	5.0	10.1	5.7	9.1	27.6	4.5	5.2	4.4	6.5	8.8	26.2
ky	1.9	247.3	1.2	129.5	47.6	12.2	9.8	14.6	7.6	12.9	34.1	4.3	4.7	4.1	6.5	8.6	26.2
lb	1.1	150.1	1.2	130.2	13.2	4.7	5.2	10.8	5.8	9.3	28.0	4.3	5.1	4.1	6.5	8.7	26.0
lo	2.7	356.5	1.2	125.9	64.7	30.8	19.6	22.7	9.3	17.5	44.9	4.6	6.1	5.2	6.5	8.6	26.4
lt	1.1	137.6	1.2	125.8	8.5	4.4	4.7	9.2	5.9	8.7	27.3	4.5	5.1	4.2	6.4	8.6	27.6
lv	1.1	144.9	1.2	126.1	13.0	4.8	5.0	9.6	5.8	9.2	27.9	4.6	5.0	4.3	6.7	8.5	26.9
mg	1.3	163.7	1.3	142.4	13.0	5.5	5.7	11.1	6.2	9.8	28.8	4.6	6.0	4.8	6.9	9.0	26.6
mi	1.2	152.0	1.3	140.0	7.9	4.8	5.3	10.7	5.8	9.9	28.2	4.7	5.7	4.6	6.6	9.0	26.8
mk	1.9	248.2	1.3	137.7	44.5	12.2	9.8	14.7	7.6	12.9	37.7	4.3	4.9	3.9	6.8	8.8	27.3
ml	3.1	406.9	1.4	148.4	63.5	37.6	21.0	25.9	17.9	19.4	46.7	5.4	7.8	6.5	7.0	9.2	26.6
mn	1.9	249.0	1.3	139.7	43.9	12.2	10.3	15.0	7.5	13.5	33.5	4.9	5.3	4.7	6.6	8.9	26.4
mr	2.7	351.5	1.3	140.2	60.1	26.8	17.1	22.9	9.5	17.0	42.1	5.1	6.5	5.0	6.9	9.0	26.6
ms	1.1	144.9	1.1	124.5	14.0	4.2	4.4	9.7	5.9	8.9	27.9	3.6	4.3	3.6	6.6	8.3	26.0
mt	1.2	152.0	1.2	127.2	16.3	5.2	5.5	11.0	5.7	9.9	31.6	4.4	5.1	4.3	6.5	8.6	33.1
my	3.5	460.0	1.2	136.1	70.4	31.9	21.3	29.7	11.6	21.7	51.4	5.1	6.9	6.3	7.0	8.9	26.4
ne	2.6	335.4	1.2	130.3	61.2	24.2	16.1	22.0	9.1	16.0	40.7	4.6	5.5	4.3	7.5	8.7	26.2
nl	1.1	146.0	1.1	125.5	14.1	4.1	4.8	9.9	6.0	9.2	27.8	3.8	4.5	3.5	6.4	8.5	25.9
no	1.0	133.4	1.1	115.7	13.3	3.8	4.4	9.1	5.7	9.0	27.4	3.5	4.0	3.3	6.6	8.2	25.9
ny	1.1	145.8	1.1	121.6	16.6	4.7	4.8	10.3	5.4	9.6	27.1	3.9	4.5	3.9	6.4	8.4	25.7
pa	2.6	340.1	1.6	179.3	47.3	26.9	17.9	23.8	9.2	16.7	40.8	7.4	12.6	8.6	7.3	10.5	28.5
pl	1.1	146.5	1.2	129.0	11.9	4.7	5.1	10.1	6.0	9.4	32.3	4.5	5.3	4.3	6.5	8.4	25.8
ps	1.6	212.3	1.3	145.4	31.5	8.4	7.7	14.7	6.9	11.9	31.1	4.6	6.3	5.0	6.8	9.1	27.1
pt	1.1	145.8	1.1	124.0	14.9	4.1	4.4	9.3	5.9	9.4	28.1	3.8	4.3	3.3	6.4	8.4	25.7
ro	1.2	155.1	1.2	135.3	12.8	4.8	5.1	10.4	6.0	9.5	28.5	4.5	5.4	4.3	6.9	8.8	26.3
ru	2.0	257.3	1.3	142.5	44.6	12.5	9.9	14.5	7.9	13.0	34.5	4.6	5.6	3.8	6.6	9.0	26.6
sd	1.6	209.4	1.3	145.2	30.7	8.1	7.5	14.8	6.7	12.0	30.9	4.8	6.5	5.3	6.8	9.2	27.1
si	2.6	342.0	1.4	149.2	56.4	28.0	18.2	24.0	9.2	18.1	40.9	5.7	7.5	6.4	6.8	9.3	27.0
sk	1.1	142.0	1.1	124.7	12.2	4.5	4.9	9.6	5.9	8.8	27.6	4.5	4.9	4.1	6.3	8.4	23.2
sl	1.0	132.4	1.1	117.5	11.2	3.9	4.5	9.0	5.8	8.8	27.7	3.9	4.3	3.5	6.6	8.3	27.3
sm	1.2	158.2	1.3	146.3	7.5	4.9	5.6	11.4	5.8	10.1	28.2	4.5	6.2	4.8	6.7	9.2	26.2
sn	1.1	145.9	1.1	124.1	14.9	4.7	4.7	10.2	5.7	9.9	27.8	4.1	4.9	4.1	6.5	8.5	26.5
so	1.2	150.1	1.2	133.5	11.1	4.8	5.1	11.0	5.7	9.8	27.8	4.4	5.4	4.5	6.5	13.2	26.7
sq	1.2	156.6	1.2	134.6	14.1	5.4	5.5	10.8	6.0	9.6	28.5	4.7	5.5	4.6	6.8	8.8	26.3
sr	1.8	235.2	1.2	136.0	42.2	11.1	9.2	13.8	7.3	12.5	32.9	4.3	4.9	4.0	6.8	8.8	26.3
st	1.2	157.2	1.3	142.9	9.1	5.2	5.4	11.5	5.6	10.0	27.8	4.5	6.0	4.8	6.7	9.1	26.4
su	1.0	136.6	1.1	118.0	13.6	3.9	4.3	9.2	5.6	8.9	27.4	3.6	4.2	3.5	6.3	8.3	25.3
sv	1.0	135.8	1.1	114.9	15.4	3.9	4.4	9.2	6.1	8.7	27.2	3.7	4.1	3.3	6.3	8.3	26.6
sw	1.0	136.7	1.1	121.8	10.8	4.1	4.5	9.6	5.7	8.9	27.5	3.8	4.4	3.6	6.7	8.1	25.9
ta	3.2	416.6	1.4	153.2	63.2	39.4	22.0	24.9	10.7	19.8	47.6	5.5	8.2	6.7	7.1	9.4	28.2
te	2.7	349.5	1.3	140.3	59.9	30.0	17.9	22.2	9.4	16.9	41.9	4.8	6.5	5.6	6.9	9.0	26.5
tg	2.0	262.4	1.4	150.4	42.7	14.2	11.0	16.4	7.8	13.4	34.7	5.3	6.5	5.3	7.0	9.2	26.6
th	2.8	360.8	1.2	134.4	62.7	30.5	19.2	20.5	9.8	17.5	42.6	4.6	6.8	5.1	6.8	8.8	27.7
tr	1.1	146.2	1.1	124.2	15.0	4.8	4.8	9.5	5.7	9.4	28.1	4.2	4.8	4.1	6.7	8.4	26.0
uk	1.9	243.0	1.3	140.9	42.0	11.7	9.6	14.2	7.6	12.8	33.5	4.7	5.4	4.3	6.5	8.9	26.5
ur	1.8	229.0	1.4	150.8	34.1	10.1	8.7	15.7	7.4	12.3	32.7	4.9	7.1	5.4	7.0	9.4	28.2
uz	1.1	147.7	1.1	122.7	16.9	4.8	5.1	10.7	6.1	9.0	27.9	4.1	4.8	4.0	6.8	8.4	23.2
vi	1.4	181.4	1.6	179.6	1.0	7.1	6.1	11.7	6.5	10.6	29.5	7.3	12.2	8.3	6.9	10.2	28.3
xh	1.1	137.6	1.1	115.0	16.4	4.4	4.5	9.9	5.5	9.4	30.2	3.9	4.4	3.8	6.3	8.2	25.6
yi	1.9	253.8	1.3	145.6	42.6	12.5	13.2	18.1	7.4	13.5	34.1	5.0	6.6	5.7	6.7	9.2	27.2
yo	1.3	166.9	1.3	144.8	13.2	6.3	6.3	11.9	6.2	10.0	29.0	5.3	6.8	5.4	6.9	9.3	26.9
zh	0.9	119.4	1.1	117.5	1.6	3.4	4.0	6.0	5.6	8.6	27.2	3.4	4.9	3.3	6.3	8.2	25.4
zu	1.1	146.8	1.1	121.3	17.4	4.8	4.8	10.6	5.7	9.9	27.7	4.1	4.7	4.1	6.4	8.4	26.5

Table 4: Results for each of the analyzed languages. The left-hand columns contain the comparison of enoding lengths *UTF-8* and MYTE. The right-hand columns present performance (BPEB) and inference time of corresponding language models ByT5 and MyT5. All numbers are averages across the FLORES-200 test split.