Does Subword Vocabulary hold back Machine Translation?

Anonymous ACL submission

Abstract

Subword tokenization is a heuristic to find contiguous pieces of characters that occur frequently, e.g., prefixes (dis-) and suffixes (-ing). However, natural language includes many more diverse patterns involving longer range dependencies, e.g., non-concatenative morphology in Arabic (Figure 1). A more expressive method to find such dependencies is to learn a vector-quantized codebook of tokens from raw bytes. We evaluate such learnt tokenizers on the task of machine translation across six language pairs and find that while they do not outperform subwords in general, they are more robust to misspellings and better on very short and very long sentences (by as much as 70%). We also demonstrate why they have a preference for representing non-concatenative morphologies.

1 Introduction

Byte Pair Encoding (Sennrich et al., 2016), the default method used in most language models, starts with a vocabulary of only the 256 possible bytes and repeatedly merges the tokens that occur most frequently next to each other (e.g., $t + h \to th; th + e \to the; \ldots$). The vocabulary of GPT-4, for instance, is obtained after 100,000 such merges, leading to some arguably unnecessary tokens like .translatesAutoresizingMaskIntoConstraints , //————————————————————————————\n\n , and abcdefghijklmnopqrstuvwxyz [1].

Recent work has shown countless limitations with BPE subwords. Technical domains such as biomedical documents (Boecking et al., 2022a), source code (Dagan et al., 2024), and financial articles (Thawani et al., 2023b) benefit from pretraining their own tokenizer for improved language understanding.

[1] Source of GPT-4 vocabulary: https://gist.github.com/s-macke/ae83f6afb89794350f8d9a1ad8a09193

كتب	k-t-b	"write" (root form)
كَتَبَ	**kataba**	"he wrote"
كَتَّبَ	**kattaba**	"he made (someone) write"
إكْتَتَبَ	**iktataba**	"he signed up"

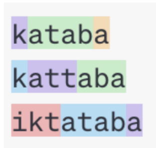

Figure 1: Left: Non-concatenative morphology in Arabic often interleaves letters within the root (Clark et al., 2022). Right: Subword tokenization in GPT-4 instead only captures 'contiguous' sequences of characters.

Another key dimension where subwords lack is language inclusivity (Team et al., 2022). Chinese characters, for instance, can be often represented better at the stroke level (Si et al., 2023). On the other hand, non-concatenative languages like Arabic can benefit from capturing long-range dependencies and not only contiguous patterns in characters - as seen in Figure 1.

The research community has proposed several alternative tokenizers to improve NLP models (Thawani et al., 2023a; Clark et al., 2022; Kumar and Thawani, 2022; Fleshman and Durme, 2023). However, each of these tokenizers also modifies the model architecture, number of parameters, vocabulary size, and/or the training corpus, thereby confounding the benefits of *only* the tokenizer vocabulary (see Table 1).

This paper studies the effects of switching to a more expressive tokenizer while controlling for all the above confounders, in the context of neural machine translation.

Our preferred alternative to subwords is a codebook learnt using vector quantization when autoencoding words in different languages (Samuel and Øvrelid, 2023) . It is a lossless arrangement of the vocabulary space that does not merely segment character sequences on the surface level, instead learns longer range dependencies among the constituent characters. We borrow the intermediate Factorizer tokenization depicted in Figure 2 and

Tokenizer	Citation	Architecture	Vocab Size	Parameters	Train Data
FastText	Bojanowski et al. (2017)	No	No	No	No
ELMo	Peters et al. (2018)	No	No	No	No
CharBERT	El Boukkouri et al. (2020)	Yes	No	No	Yes
CharFormer	Tay et al. (2021)	No	No	Yes	Yes
LOBEF	Sreedhar et al. (2022)	No	No	No	Yes
CANINE	Clark et al. (2022)	No	No	No	Yes
ByT5	Xue et al. (2022)	No	No	Yes	Yes
MegaByte	Yu et al. (2023)	No	No	No	Yes
RetVec	Bursztein et al. (2023)	No	No	No	Yes
eByte/eChar	Thawani et al. (2023a)	No	No	Yes	Yes
Factorizer	Samuel and Øvrelid (2023)	Yes	Yes	Yes	Yes

Table 1: Literature Review of alternative tokenizers and what they control for. We work with Factorizer, the only tokenizer that controls for all dimensions and makes it possible to compare directly against a subword vocabulary.

described in Section 3.

We acknowledge that codebook-learned tokenizers have several shortcomings. They are not as directly interpetible as subwords. They require training from scratch since most pretrained language models today use subword vocabularies instead. They lack the inductive bias that characters appearing close may form coherent units, which limits expressivity but is nonetheless a useful bias (Cao, 2023).

Nevertheless, we believe our empirical and controlled analysis of their performance in machine translation offers several contributions:

1. We are the first to compare BPE tokenizers to a learnt vocabulary with the same size and the same architecture on the downstream task of Neural Machine Translation.

2. We show that while BPE outperforms Factorizer in general, the latter is more robust to noise and for very short and very long sentences (outperforms by as much as 70%).

3. We analyze why Factorizer prefers non-concatenative morphologies like Arabic.

We will publicly release all code (see supplementary material) and checkpoints upon acceptance.

2 Background

Here, we describe the key tokenization strategies that we compare without modifying the underlying model architecture in any way. We refer the interested reader to Mielke et al. (2021) for a deeper survey on tokenization in NLP.

2.1 Bytes

Most natural language text on the internet is encoded using UTF-8 byte encodings, therefore a byte-level representation of text makes for a convenient option. Their vocabulary size is restricted to a mere 256 possible bytes, and most Latin languages require a single byte per character.

Such approaches (Xue et al., 2022; El Boukkouri et al., 2020), however, suffer from being slow to infer due to large description lengths, particularly on non-Latin scripts (Edman et al., 2023).

2.2 Byte Pair Encoding

The modern workhorse of tokenization in NLP is a heuristic atop byte representations called Byte Pair Encoding. Starting from a base of 256 bytes and a training corpus, the most frequently occurring byte pairs are incrementally merged, e.g., t+h →th, th+e→the, and so on.

Nearly all large language models today (Touvron et al., 2023a,b; Groeneveld et al., 2024; Jiang et al., 2023) rely on Byte Pair Encoding as their base tokenizer, with different number of merges. GPT3 (Brown et al., 2020) uses a vocabulary of 50,257 BPE tokens (50,000 merges and a special token) while GPT4 (OpenAI et al., 2023) pushes it further to 100,000 merges.

One of the main goals of this paper is to control for dimensions like vocabulary size, hence we train our own BPE on the training set of each dataset (independently for source and target sides) with a final size of 794 BPE tokens - the same as the factorizer (see next section).

3 Methodology

We reuse the Factorized Subword Encoding Samuel and Øvrelid (2023), which trains an au-

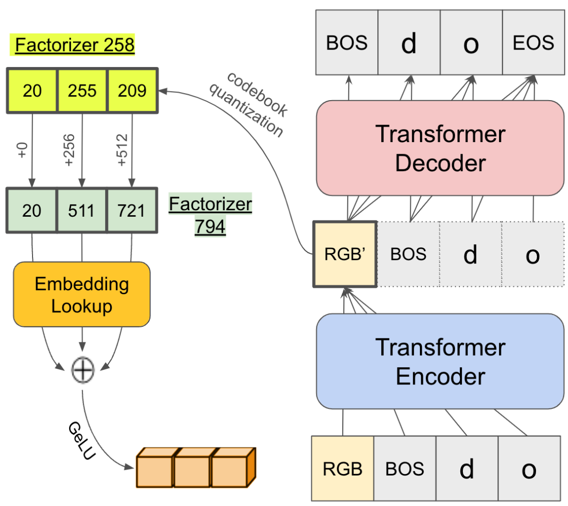

Figure 2: Pictorial depiction of how the Factorizer (Samuel and Øvrelid, 2023) learns token embeddings as an autoencoder (seen here reconstructing the word 'do') where the final summed embeddings of the word are used to evaluate on syntactic tasks. We specifically borrow these intermediate codes labelled Factorizer 258 and Factorizer 794 in our paper as stand-in replacements for a BPE tokenizer, enabling fair comparison on NMT.

toencoder to learn to decompose subwords into triplet codes, each ranging from $0 - 255$, resembling an RGB color code[2]. Such a factorization helps construct tokens with compositional units, e.g., *melon* is represented as $[30, 255, 209]$, *melons* as $[261, 255, 209]$ and *watermelons* as $[208, 235, 109]$, $[45, 255, 209]$, sharing most of their encoding. We refer the interested reader to the original paper for more implementation and training details, which we summarize in Figure 2.

They focus on pooling these RGB embeddings to give a single vector representation per subword, and then use them in a BERT-style model for morpho-syntactic tasks. We merely borrow their autoencoding codebook to discretize text in the same way as a BPE tokenizer would. Their original vocabulary size is 256 x 3 (one each for RGB) equivalent to 768 unique tokens.

Another alternative we try is to keep the vocabulary size 256 and let the model's positional encodings learn patterns that inform whether a given code represents the R, the G, or the B part of a token's representation.

We use both variants in our experiments, distinguishing them by the size of their vocabulary as Factorizer 794 and Factorizer 258[3]. They correspond nearly perfectly to the vocabulary sizes of our baselines: BPE (794) and Bytes (256).

4 Experiment Setup

Our primary research question is to evaluate a learnt Factorizer vocabulary with BPE subwords. We operationalize this in the form of a neural machine translation experiment to compare different tokenizers where the same model is trained from scratch on the same dataset for the same number of epochs with the same optimizer configuration.

Model Our base model is a 6 layer transformer encoder-decoder (Vaswani et al., 2017) that has 8 attention heads, 512 hidden vector units, and a feed forward intermediate size of 2048, with GeLU activation (Hendrycks and Gimpel, 2023). We use label smoothing at 0.1, and a dropout rate of 0.1. We use the RTG [4] library for model implementation and an extended version of NLCodec library (Gowda et al., 2021) for tokenization.

Datasets: We use a variety of machine translation datasets in our experiments, preprocessed with the Moses tokenizer (Koehn et al., 2007). For each language pair, we summarize our training, development, and test sets in Table 2, each based on the following source:

1. **Europarl Corpus**: Originating from the European Parliament proceedings, this multilingual dataset is focused on political and legislative language (Koehn, 2005).

2. **News Commentary Corpus**: This corpus includes multilingual news commentary articles, with exposure to current events and journalistic language (Tiedemann, 2009).

3. **WMT Newstest Sets**: Part of the annual Workshop on Machine Translation evaluation, these news article sets are used for benchmarking translation system performance (Kocmi et al., 2022).

4. **Flores Benchmark**: Designed for evaluating translation in low-resource languages, Flores includes a broad domain range, improving model versatility (NLLB Team et al., 2022).

[2] Unlike the RGB continuous spectrum, here $[0, 1, 2]$ may have more in common with $[39, 40, 41]$ than with $[1, 2, 3]$.

[3] Corresponding to 768 and 256 respectively but with a few additional special tokens to denote $[BOS]$, $[EOS]$, etc.

[4] https://github.com/isi-nlp/rtg

Language Pair	Dataset	Type	Versions	# Sentences	Size (MBs)	# Chars/Sentence
French-English (Fr-En)	Europarl News Commentary	Training Training	v7 v16	2,002,756 365,510	647.69 116.05	Fr-166.69; En-147.66
	Newstest	Development	2010	2,489	0.71	Fr-147.53 ; En-130.88
	Newstest Newstest Newstest	Test Test Test	2011 2012 2013	3,003 3,003 3,003	0.85 0.82 0.72	Fr-141.48 ; En-126.0 Fr-146.67 ; En-131.06 Fr-126.41 ; En-109.98
German-English (De-En)	Europarl News Commentary	Training Training	v10 v16	1,817,758 388,482	585.08 120.34	De-167.45 ; En-147.06
	Newstest	Development	2017	3,004	0.71	De-122.04 ; En-111.07
	Newstest Newstest Newstest	Test Test Test	2018 2019 2020	2,998 2,000 785	0.74 0.43 0.43	De-107.27 ; En-101.98 De-126.66 ; En-116.22 De-282.84 ; En-263.92
Spanish-English (Es-En)	Europarl News Commentary	Training Training	v7 v16	1,960,641 369,540	619.08 114.09	Es-161.68 ; En-147.58
	Newstest	Development	2010	2,489	0.69	Es-142.36 ; En-130.88
	Newstest Newstest Newstest	Test Test Test	2011 2012 2013	3,003 3,003 3,003	0.83 0.81 0.71	Es-140.73 ; En-131.06 Es-123.09 ; En-109.98 Es-138.57 ; En-126.0
English-Arabic (En-Ar)	Flores200 News Commentary	Training Training	v1 v16	997 140,929	0.33 132.74	En-289.44 ; Ar - 353.62
	UN Test	Development	v1	4,000	1.79	En-175.36 ; Ar - 148.38
	Flores200 devtest	Test	v1	1,012	0.34	En-130.4 ; Ar-114.93
Spanish-Arabic (Es-Ar)	Flores200 News Commentary	Training Training	v1 v16	997 132,616	0.36 130.82	Es-335.49 ; Ar-351.81
	UN Test	Development	v1	4,000	1.9	Es-200.63 ; Ar-148.38
	Flores200 devtest	Test	v1	1,012	0.37	Es-155.14 ; Ar-114.93
French-Arabic (Fr-Ar)	Flores200 News Commentary	Training Training	v1 v16	997 104009	0.35 105.57	Fr-345.85 ; Ar-354.56
	UN Test	Development	v1	4000	1.91	Fr-198.43 ; Ar-148.38
	Flores200 devtest	Test	v1	1012	0.38	Fr-155.77 ; Ar-114.93

Table 2: Summary of our Training, Development, and Test Datasets on six language pairs.

5. **United Nations (UN) Test Sets**: Derived from official UN documents, this dataset introduces models to complex diplomatic and international terminology (Ziemski et al., 2016).

Training and Evaluation We use the Adam optimizer (Kingma and Ba, 2017) with a controlled learning rate that warms up for 16K steps followed by a decay rate recommended for training transformer models. Each model is trained from scratch, and the hyperparameters (per language pair) are chosen by grid search to optimize the baseline validation BLEU. We train all models for up to $100,000$ steps (early stop by development loss with a patience of 5) with batch size $24,000$. We report sacreBLEU (Post, 2018) and chrF ($\beta = 2$) scores (Popović, 2015).

As is common in machine translation experiments, our models do not share source and target vocabularies. In most experiments below, we further isolate the effects of tokenization to a single side (source or target) while fixing the other side to be the default baseline with $8,000$ BPE tokens. Doing so at the target side has the added advantage that the autoregressive decoding speed at inference is unaffected by the source vocabulary, which is one of the prominent critiques against, say, byte-level models.

5 Results and Discussion

The purpose of this work is to compare traditionally used tokenizers like Byte and BPE subwords to the learnt tokenizers: Factorizer 258 and Factorizer 794. We break down our results into the following research questions:

5.1 How well do learnt tokenizers *encode* source text and *decode* target text?

We first experiment with different source-side tokenizers while keeping the target side as BPE 8K. Table 3 shows that Factorizer (794) does not outperform BPE but is better than Bytes when translating Arabic to other languages. We theorize that the Bytes tokenizer does relatively better on English primarily due to how UTF-8 encodes each Latin alphabet with a single byte each, whereas Arabic alphabets require two bytes each.

Based on the above results, we further experiment with the two best tokenizers BPE 794 and Factorizer 794 at target-side in machine translation. The smaller vocabulary Byte and Factorizer 258

	Factorizer 794		BPE 794		Byte 258		Factorizer 258	
	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF
$En \rightarrow De$	22.4 ± 4.4	53.4 ± 3.0	22.7 ± 4.6	54.4 ± 3.2	25.2 ± 5.2	55.6 ± 3.4	20.8 ± 4.0	52.2 ± 2.9
$En \rightarrow Fr$	22.4 ± 0.7	53.7 ± 1.0	21.6 ± 2.2	53.1 ± 2.3	25.1 ± 0.7	56.0 ± 0.9	24.0 ± 0.7	52.7 ± 1.0
$En \rightarrow Es$	28.0 ± 1.5	54.8 ± 1.3	29.3 ± 1.5	56.2 ± 1.3	32.1 ± 1.8	56.9 ± 1.6	27.9 ± 1.5	54.1 ± 1.3
$En \rightarrow xx$	24.3	54.0	24.5	54.6	**27.5**	**56.1**	24.2	53.0
$Ar \rightarrow En$	20.5 ± 0.3	48.5 ± 0.3	22.2 ± 0.1	49.8 ± 0.5	21.2 ± 0.7	48.2 ± 0.3	17.7 ± 0.1	45.0 ± 0.2
$Ar \rightarrow Fr$	13.9 ± 0.5	42.4 ± 0.1	15.0 ± 0.3	44.1 ± 0.1	11.2 ± 0.8	38.7 ± 0.7	11.1 ± 0.1	38.6 ± 0.1
$Ar \rightarrow Es$	12.6 ± 0.3	39.7 ± 0.3	13.2 ± 0.1	40.9 ± 0.1	4.9 ± 3.3	27.3 ± 6.2	10.5 ± 0.2	37.4 ± 0.2
$Ar \rightarrow xx$	15.7	43.6	**16.8**	**44.9**	12.4	38.1	13.1	40.3

Table 3: Comparison of different source tokenizers with the target fixed (xx → BPE-8K) across 6 language pairs, along with standard deviations over 3 runs with different random seeds. English source experiments are averaged over three different test sets, resulting in higher variance. We also report (micro) averages grouped by source language. Takeaway: Factorizer does not outperform BPE but is better than Bytes when translating Arabic.

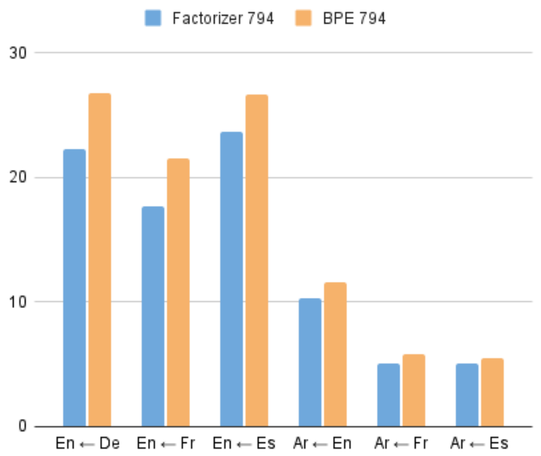

Figure 3: BLEU scores on target side with the source side fixed as (xx ← BPE-8K) across six language pairs. BPE consistently outperforms Factorizer.

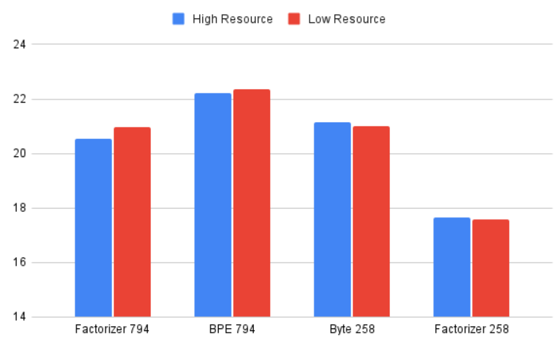

Figure 4: Data Scarcity: BLEU scores over Ar → En with different source-side tokenizers (target-side fixed at BPE 8k). Most tokenizers lose performance in a low resource setting but Factorizer 794 gains the most.

tokenizers are also particularly slow at inference, since they must autoregressively decode more number of times for the same sentence than BPE 794 and Factorizer 794. Figure 3 shows again that while Factorizer performs competitively with BPE, it is unable to beat it for any of the six language pairs.

In the following sections, we perform further ablations primarily on the Arabic-English translation task, since Factorizer shows relative promise in encoding Arabic. Moreover, the Ar → En task helps us qualitatively analyze model outputs in English (Section 5.4).

5.2 How robust are tokenizers to data scarcity?

Prior work (Samuel and Øvrelid, 2023) has shown the benefits that alternative tokenizers have when training with low resources. Here, we evaluate the relative drop in performance of our models when trained on lower resources.

More specifically, we experiment with Arabic → English translation where the training set is now UN Test (4,000 examples) and the development set is Flores 200 (997 examples). In the high resource setting, the total training set had 141,926 examples and the development set had 4,000 examples. For fair comparison, our test set in both settings is Flores 200 devtest (1,012 examples).

Figure 4 reports BLEU scores when comparing different source-side tokenizers, keeping target-side tokenizer fixed at our default BPE 8k. We find that while most tokenizers lose some score in the low resource setting, Factorizer 794 on the contrary gains the most, demonstrating better robustness to data scarcity.

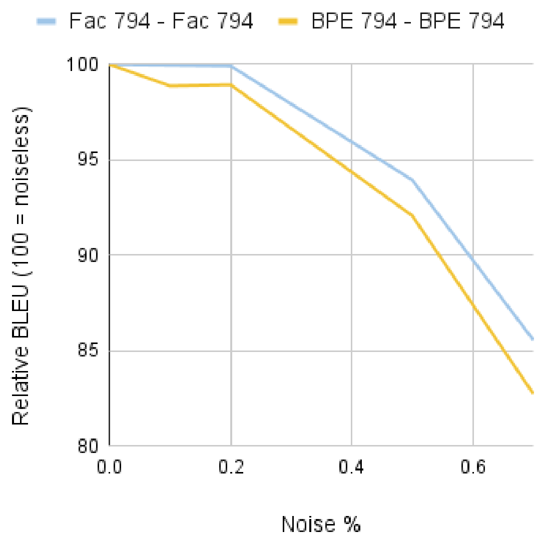

Figure 5: Ar→En relative BLEU scores (100 denotes noiseless[5]) with varying degrees of noise added to the test source sentences. Factorizer performance relatively degrades less than BPE as noise increases.

5.3 How robust are tokenizers to noise?

Following Samuel and Øvrelid (2023) we experiment with adding different degrees of artificial noise in our Arabic→English experiments with BPE 794-BPE 794 and Factorizer 794-Factorizer 794 [5]. We add, remove, or replace each non-space character with a certain probability in the test set source sentences (Arabic); the training set remains uncorrupted in each case. In line with previous work, Figure 5 find that Factorizer performance relatively degrades less than BPE as noise increases.

5.4 Do different tokenizers specialize in different kinds of translations?

We note in Table 3 how Byte-tokenized models work better for Latin scripts than non-Latin ones. This can be possibly explained by the inherent bias within UTF-8 encoding scheme which yields a single byte to all Latin characters but as many as three bytes per character for languages that appear later in the Basic Multilingual Plane (BMP).

Here, we ask similarly what other factors may influence the performance of a tokenizer in machine translation. We use the Compare-MT (Neubig et al., 2019) library to stratify results according to source length, target length, frequency of words, presence of key phrases, and other dimensions.

[5]The noiseless BLEU scores are respectively 23.4 and 20.1 (in line with above results).

Length	Factorizer-794	BPE-794
<10	**17.33**	10.73
[10,20)	15.06	**16.65**
[20,30)	17.45	**18.63**
[30,40)	**20.22**	19.30
[40,50)	18.62	**19.43**
[50,60)	17.98	**19.58**
>=60	**45.30**	33.16

Table 4: BLEU scores on Arabic → English stratified by lengths. Factorizer particularly outperforms when the reference is either very short or very long.

Table 4 depicts a stratification by length of target reference. We find that Factorizer significantly outperforms BPE on very short and very long translations, by as much as 70%. Table 5 also highlights such representative samples from the test set of our Arabic → English experiments.

5.5 Can we quantify the morphological preference of tokenizers?

Our experiments show that relatively, Factorizers perform better on Arabic than say, English. We note in Figure 1 how the non-concatenative morphology of Arabic may be a factor behind this result. In this subsection, we further quantify this intuition.

We test the hypothesis of whether BPE and Factorizer are separately suited to be better at different kinds of morphologies. To this end, we cluster the top 10,000 words in both Arabic and English by their root form (Sylak-Glassman, 2016; van der Zwaan et al., 2019), e.g., the root form `have` maps to the following common words: `have, has, had, having`. Next, we tokenize each such word using the two tokenizers (BPE 794 and Factorizer 794), and count the subset of encoding that is 'most representative' of the root cluster.

We define representativeness here as the fraction of words that share this code within this cluster. For example, if two of the above four forms of the root `have` include a code `ha##` and six other English words also include this code, then the representativeness score for this cluster in BPE is $\frac{2}{8} = 0.25$.

We plot the histograms of representativeness scores over 1,410 English roots and 73 Arabic ones in Figures 6 and 7. Distributions that are shifted towards the right side on the X-axis indicate a more representative code that captures root forms. We observe that while BPE subwords are better suited

	Text	SentBLEU
Reference	The harbor was the site of an infamous naval standoff in 1889 when seven ships from Germany, the US, and Britain refused to leave the harbor.	
Factorizer	The facility was the site of a notorious sea-lane confrontation in a little-noticed year when seven ships from Germany, the US, and Britain refused to leave the air.	55.20
BPE	Seven ships from Germany, the United States, and Britain refused to leave.	14.94
Reference	The Internet combines elements of both mass and interpersonal communication.	
Factorizer	The Internet combines elements of both mass and private communication.	80.50
BPE	The Internet brings together elements of both public and personal communication.	26.78
Reference	Argentina is well known for having one of the best polo teams and players in the world.	
Factorizer	Argentina is famous for having one of the best teams and Buddhist players in the world.	52.86
BPE	Argentina is notorious for the existence of one of the world ' s best statesmen.	17.40
Reference	Christmas is one of the most important holidays of Christianity, and is celebrated as the birthday of Jesus.	
Factorizer	Christmas is one of Christianity ' s most important Christmas habits, celebrated as Christmas.	23.41
BPE	Christmas is one of the most important holidays of Christianity, and is celebrated as Christmas 's birthday.	76.83
Reference	As knowledge of Greek declined, the West found itself cut off from its Greek philosophical and scientific roots.	
Factorizer	While knowledge has declined in Greeks, the West has found itself insulated from its philosophical roots and Greek science.	13.80
BPE	As Greek knowledge declined, the West found itself isolated from its philosophical and scientific roots.	42.68
Reference	A couple may decide it is not in their best interest, or in the interest of their child, to raise a baby.	
Factorizer	She may decide that she is neither good nor in her child ' s interest to rank a baby.	10.37
BPE	uan may decide that it is not in their interest, or in the interest of their child, to have a baby.	60.26

Table 5: Representative samples of Arabic → English translations - three examples each of where Factorizer significantly outperforms BPE and vice versa (as measured by Sentence BLEU). We highlight the winning system's successes and failures.

to the concatenative morphology of English, Arabic root forms that share non-concatenative morphological features are better encapsulated by the learnt codes in Factorizer (blue distribution leans more to the right, i.e., higher representativeness).

6 Related Work

Some recent work has challenged subword tokenization schemes. Table 1 highlights the different kinds of alternative tokenizations existing in prior work and why this paper works with the Factorizer, the only tokenizer that controls for all dimensions and makes it possible to compare directly against a subword vocabulary. This section summarizes the different efforts by the community towards alterantive tokenization:

Character/Byte-level ByT5 (Xue et al., 2022), CANINE (Clark et al., 2022), and SubChar (Si et al., 2021) propose using very small fixed-length units such as characters, bytes, or glyph strokes instead of dynamic-length subwords or words. This often comes at the expense of larger sequence lengths and more compute requirements, especially for a transformer architecture which typically has a complexity of $\mathcal{O}(n^2)$ in number of input tokens. Edman et al. (2023) investigate byte and subword-level models for machine translation.

Beyond word level CodeBPE (Chirkova and Troshin, 2022) and Multi Word Expressions (Kumar and Thawani, 2022; Zaninello and Birch, 2020; Rikters and Bojar, 2017) show promise in yet larger tokens that cross word boundaries, e.g., a vocabulary with single tokens for the strings "for i in range" or "New York City" respectively.

Learnt subword segmentation Some methods (Mofijul Islam et al., 2022; Kaushal and Mahowald, 2022; Pinter et al., 2021; Tay et al., 2021; Provilkov et al., 2020; Wang et al., 2021) parameterize the process of segmentation by pooling character n-grams or sampling one of the many ways to segment a given word. In contrast, we are interested in a different rearrangement of the vocabulary that does not segment words at the surface level alone.

Domain specific tokenization Several domains have benefited from a custom tokenization strategy (Dagan et al., 2024). Numbers are often inconsistently segmented into subwords, leading to decreased arithmetic (Wallace et al., 2019) and estimation (Thawani et al., 2021) skills. The extent of these numeric limitations is so dire that GPT-4 (OpenAI et al., 2023) has an explicit workaround

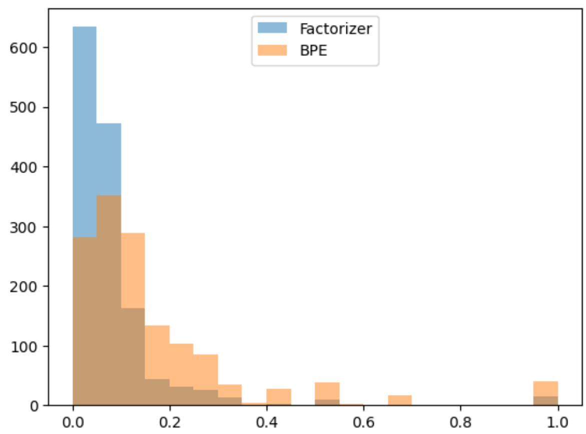

Figure 6: Representativeness in English. BPE 794 codes well represent more root forms than Factorizer 794 (rightwards is better). See Section 5.5 for details.

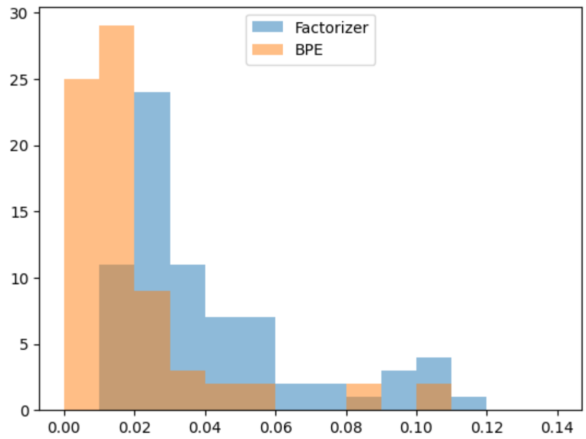

Figure 7: Representativeness in Arabic. Factorizer 794 codes well represent more root forms than BPE 794 (rightwards is better). See Section 5.5 for details.

of adding all numbers from 0 to 999 as individual tokens to the model's vocabulary. Boecking et al. (2022b) train a better tokenizer for the biomedical domain and Dagan et al. (2024) perform a similar analysis over code language models.

7 Conclusion

In conclusion, our study explored the impact of tokenization schemes on neural machine translation performance by comparing traditional Byte Pair Encoding (BPE) with a recent, learned tokenizer known as Factorizer. Our experiments, conducted across six language pairs, revealed that while BPE continues to hold its ground as the superior tokenizer in most scenarios, Factorizer shows promise, particularly when translating from Arabic. Notably, Factorizer outperformed BPE in translating very short and very long sentences, indicating its potential in handling edge cases effectively.

We rigorously analyze one of the factors influencing this relative preference for BPE towards inflectional morphologies like English and Factorizer towards non-concatenative morphologies like Arabic. We find that learnt codebooks better represent the non-concatenative root forms in Arabic than subword heuristics (Figure 7).

Our findings underscore the importance of continuing to explore and refine tokenization techniques in the field of neural machine translation. While BPE remains a strong baseline, the potential for improvement with learned tokenizers like Factorizer warrants further investigation, particularly in language pairs and scenarios where traditional methods may falter.

8 Limitations

We acknowledge that codebook-learned tokenizers have several shortcomings. They are not as directly interpetible as subwords. They need to be trained on a corpus (though so do subword tokenizers), and cannot be plugged into a pretrained language model. They lack the inductive bias that characters appearing close may form coherent units.

Our paper empirically analyses the research question: to what extent could BPE tokenizers be inhibiting machine translation? While our results indicate that Factorizers (codebook-learnt tokenizers) do not outperform subword-based models in general, our work highlights how and where do they perform at par.

This study is limited to machine translation, but we refer readers to the Appendix in Samuel and Øvrelid (2023) for preliminary experiments on GLUE, a general NLP benchmark. They find similarly that Factorizer does not outperform but also does not lag far behind the default BPE tokenizers.

9 Ethical Impact

We acknowledge that research on tokenization in language models is one of the fundamental steps where language diversity is essential for an equitable outcome in Generative AI.

Our work is in part an effort to evaluate tokenizers that make less assumptions about the morphology of the underlying language than BPE-like subword segmentation heuristics. We analyze in Section 5.5 how non-concatenative morphology in Arabic may influence the relatively better performance of factorizers than on English.

References

Benedikt Boecking, Naoto Usuyama, Shruthi Bannur, Daniel C Castro, Anton Schwaighofer, Stephanie Hyland, Maria Wetscherek, Tristan Naumann, Aditya Nori, Javier Alvarez-Valle, et al. 2022a. Making the most of text semantics to improve biomedical vision–language processing. In *European conference on computer vision*, pages 1–21. Springer.

Benedikt Boecking, Naoto Usuyama, Shruthi Bannur, Daniel C Castro, Anton Schwaighofer, Stephanie Hyland, Maria Wetscherek, Tristan Naumann, Aditya Nori, Javier Alvarez-Valle, et al. 2022b. Making the most of text semantics to improve biomedical vision–language processing. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, pages 1–21. Springer.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Elie Bursztein, Marina Zhang, Owen Vallis, Xinyu Jia, and Alexey Kurakin. 2023. Retvec: Resilient and efficient text vectorizer.

Kris Cao. 2023. What is the best recipe for character-level encoder-only modelling? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5924–5938, Toronto, Canada. Association for Computational Linguistics.

Nadezhda Chirkova and Sergey Troshin. 2022. Codebpe: Investigating subtokenization options for large language model pretraining on source code. In *Deep Learning for Code Workshop*.

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.

Gautier Dagan, Gabriele Synnaeve, and Baptiste Rozière. 2024. Getting the most out of your tokenizer for pre-training and domain adaptation.

Lukas Edman, Gabriele Sarti, Antonio Toral, Gertjan van Noord, and Arianna Bisazza. 2023. Are character-level translations worth the wait? comparing character- and subword-level models for machine translation.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6903–6915, Barcelona, Spain (Online). International Committee on Computational Linguistics.

William Fleshman and Benjamin Van Durme. 2023. Toucan: Token-aware character level language modeling.

Thamme Gowda, Zhao Zhang, Chris A. Mattmann, and Jonathan May. 2021. Many-to-english machine translation tools, data, and pretrained models. *CoRR*, abs/2104.00290.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Olmo: Accelerating the science of language models.

Dan Hendrycks and Kevin Gimpel. 2023. Gaussian error linear units (gelus).

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Ayush Kaushal and Kyle Mahowald. 2022. What do tokens know about their characters and how do they know it? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2487–2507, Seattle, United States. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki

Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Dipesh Kumar and Avijit Thawani. 2022. BPE beyond word boundary: How NOT to use multi word expressions in neural machine translation. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 172–179, Dublin, Ireland. Association for Computational Linguistics.

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp.

Md Mofijul Islam, Gustavo Aguilar, Pragaash Ponnusamy, Clint Solomon Mathialagan, Chengyuan Ma, and Chenlei Guo. 2022. A vocabulary-free multilingual neural tokenizer for end-to-end task learning. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 91–99, Dublin, Ireland. Association for Computational Linguistics.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, Xinyi Wang, and John Wieting. 2019. compare-mt: A tool for holistic comparison of language generation systems. *CoRR*, abs/1903.07926.

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.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Yuval Pinter, Amanda Stent, Mark Dredze, and Jacob Eisenstein. 2021. Learning to look inside: Augmenting token-based encoders with character-level information.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

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.

Matīss Rikters and Ondřej Bojar. 2017. Paying attention to multi-word expressions in neural machine translation. In *Proceedings of Machine Translation Summit XVI: Research Track*, pages 86–95, Nagoya Japan.

David Samuel and Lilja Øvrelid. 2023. Tokenization with factorized subword encoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14143–14161, Toronto, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2021. Shuowen-jiezi: Linguistically informed tokenizers for chinese language model pretraining. *arXiv preprint arXiv:2106.00400*.

Chenglei Si, Zhengyan Zhang, Yingfa Chen, Fanchao Qi, Xiaozhi Wang, Zhiyuan Liu, Yasheng Wang, Qun Liu, and Maosong Sun. 2023. Sub-Character Tokenization for Chinese Pretrained Language Models. *Transactions of the Association for Computational Linguistics*, 11:469–487.

Makesh Narsimhan Sreedhar, Xiangpeng Wan, Yu-Jie Cheng, and Junjie Hu. 2022. Local byte fusion for neural machine translation. *ArXiv*, abs/2205.11490.

John Sylak-Glassman. 2016. The composition and use of the universal morphological feature schema (unimorph schema).

Yi Tay, Vinh Q Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2021. Charformer: Fast character transformers via gradient-based subword tokenization. In *International Conference on Learning Representations*.

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.

Avijit Thawani, Saurabh Ghanekar, Xiaoyuan Zhu, and Jay Pujara. 2023a. Learn your tokens: Word-pooled tokenization for language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9883–9893.

Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro Szekely. 2021. Representing numbers in NLP: a survey and a vision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–656, Online. Association for Computational Linguistics.

Avijit Thawani, Jay Pujara, and Ashwin Kalyan. 2023b. Estimating numbers without regression. *arXiv preprint arXiv:2310.06204*.

Jörg Tiedemann. 2009. *News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces*, volume V, pages 237–248.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Janneke van der Zwaan, Dafne van Kuppevelt, Maksim Abdul Latif, Melle Lyklema, and Christian Lange. 2019. arabic-digital-humanities/root-extraction- validation-data: 0.1.0.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315, Hong Kong, China. Association for Computational Linguistics.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2021. Multi-view subword regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 473–482, Online. Association for Computational Linguistics.

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. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Lili Yu, Dániel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. Megabyte: Predicting million-byte sequences with multiscale transformers. *arXiv preprint arXiv:2305.07185*.

Andrea Zaninello and Alexandra Birch. 2020. Multiword expression aware neural machine translation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3816–3825, Marseille, France. European Language Resources Association.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).