Tokenization Impacts Multilingual Language Modeling: Assessing Vocabulary Allocation and Overlap Across Languages

Tomasz Limisiewicz and **Jiří Balhar** and **David Mareček**
Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic
`{limisiewicz, marecek}@ufal.mff.cuni.cz`

Abstract

Multilingual language models have recently gained attention as a promising solution for representing multiple languages in a single model. In this paper, we propose new criteria to evaluate the quality of lexical representation and vocabulary overlap observed in sub-word tokenizers. Our findings show that the overlap of vocabulary across languages can be actually detrimental to certain downstream tasks (POS, dependency tree labeling). In contrast, NER and sentence-level tasks (cross-lingual retrieval, NLI) benefit from sharing vocabulary. We also observe that the coverage of the language-specific tokens in the multilingual vocabulary significantly impacts the word-level tasks. Our study offers a deeper understanding of the role of tokenizers in multilingual language models and guidelines for future model developers to choose the most suitable tokenizer for their specific application before undertaking costly model pre-training.[1]

1 Introduction

Multilingual language models perform surprisingly well in a variety of NLP tasks for diverse languages (Devlin et al., 2019; Conneau and Lample, 2019; Conneau et al., 2019). It has been observed that the representation of the input sequence has a significant effect on their effectiveness (Mielke et al., 2021). In the widely used Transformer (Vaswani et al., 2017) models achieving state-of-the-art results through diverse tasks, a large fraction of parameters are allocated in the input encoding layer.[2] The popular language-independent approach to represent the input texts is to learn a vocabulary of frequently appearing strings that may consist of words or parts of words (Sennrich et al., 2016; Song et al., 2021; Kudo and Richardson, 2018).

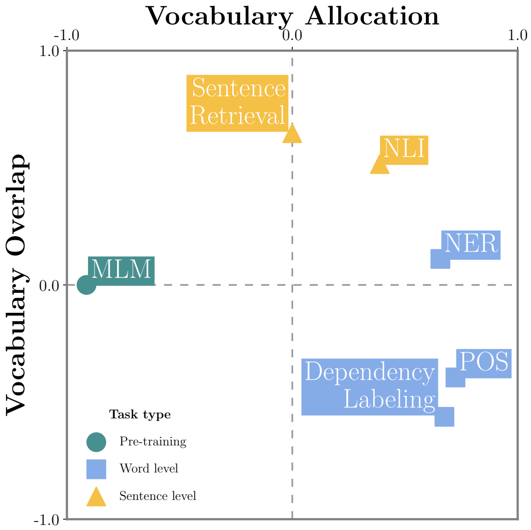


Figure 1: Mapping the impact of *vocabulary allocation* and *vocabulary overlap* on language model performance. The location of points corresponds to Spearman's correlation between vocabulary measures and the task score (see the details in Tables 3 and 5). High *vocabulary overlap* benefits NER and sentence-level tasks (NLI, sentence retrieval) and hinders POS and dependency labeling performance. High *vocabulary allocation* improves word-level tasks but leads to a decrease in masked language modeling scores. Masked language modeling is measured only in language. Thus it's unaffected by *vocabulary overlap*. Analogically, sentence retrieval is solely cross-lingual and unaffected by *vocabulary allocation*.

In this work, we focus on the characteristics of subword tokenization methods in a multilingual setting. Our main contribution is the introduction of the methods for measuring whether tokenizers effectively represent meaningful language-specific tokens in the vocabulary (*vocabulary allocation*) and whether the units they learn are shared across languages (*vocabulary overlap*). We posit the following questions:

[1] The code is available at: `github.com/tomlimi/entangled_in_scripts`.

[2] For instance, in XLM-Roberta$_{\text{Base}}$, 192M out of 270M parameters are in the input embedding layer (approximately 70%).

Findings of the Association for Computational Linguistics: ACL 2023, pages 5661–5681
July 9-14, 2023 ©2023 Association for Computational Linguistics

(Q1) How do sub-word tokenizers differ in *overlap* and *allocation* of learned vocabularies? To answer this question, we apply the metrics to tokenizers obtained with two widely used algorithms: SentencePiece Unigram LM (Kudo and Richardson, 2018), and BPE (Sennrich et al., 2016). Furthermore, we propose two methods of learning tokenizers on monolingual corpora and then combining them to allow the tokenization of multilingual texts.

(Q2) Which properties of multilingual tokenizers affect the LM's representation quality? We address this question by training small language models utilizing different tokenization methods. We evaluate the models on masked word prediction and a diverse set of downstream tasks: POS, NER tagging, dependency tree labeling, NLI, and cross-lingual sentence retrieval.

The proposed evaluation scheme offers a good prediction of language models' performance. Notably, we show that the system results significantly improve when tokenizers allocate more vocabulary units for specific languages. Our investigation shows that this aspect has a bigger influence than the *vocabulary overlap* for word-level tasks (see Figure 1). To the best of our knowledge, the interactions between multilingual *vocabulary allocation* and *vocabulary overlap* have not been investigated in past research.

2 Multilingual Subword Tokenization

The majority of the currently deployed models use subword tokenization as a way to pre-process the input texts. The input is represented as a sequence of units from a finite vocabulary, which can be translated into numeric representation by an input embedding layer.

The benefits of subword tokenization are the ability to obtain numeric representation for meaningful words frequently used in the resources and handling less frequent words by splitting them into subwords. The latter property mitigates the problem of out-of-vocabulary (OOV) words by breaking them down into smaller parts (sub-words) already present in the vocabulary. It is crucial in handling multilingual texts, especially in languages with large vocabularies and complex morphology.

In the following section, we describe two widely used algorithms of subword tokenization:

2.1 Background: Subword Tokenization

Byte-pair encoding BPE: (Sennrich et al., 2016) is a subword tokenization method that iteratively replaces the most frequent pair of vocabulary units in the input text with a single unit. The process starts with taking unique characters of the training text as the initial vocabulary. Subsequently, we take the most frequent pair of vocabulary units, merge the pair, and add it as a new unit to the vocabulary. This process is repeated until a pre-set vocabulary size N is reached.

Unigram LM: (Kudo, 2018) is the method of obtaining subword vocabulary that was first introduced as the underlying tokenizer of SentencePiece algorithm (Kudo and Richardson, 2018). The prerequisite is obtaining an extensive vocabulary, e.g., consisting of all strings present in data with at most, a predefined number of characters. The expectation-maximization algorithm is used to estimate the probability of vocabulary units. After EM convergence, the portion of units with the lowest contribution to the likelihood of the training corpus is removed from the vocabulary. The procedure is repeated until the pre-set vocabulary size is obtained.

2.2 Combining Monolingual Tokenizers

Rust et al. (2021) observed that subword tokenizers trained on monolingual data outperform multilingual ones. The latter can overrepresent the subwords specific to languages constituting a large portion of the training corpora (e.g., English). Moreover, their vocabulary is less likely to contain morphemes important in modeling low-resource languages and instead prioritizes less meaningful character sequences appearing across languages.

To alleviate this issue, we suggest utilizing monolingual tokenizers for multilingual tokenization. First, the Unigram LM tokenizers are trained on separate monolingual corpora. The tokenizers are then combined to create a tokenizer suitable for multilingual data. We propose two methods for combining monolingual tokenizers:

Language-specific Tokenization NoOverlap: We train Unigram tokenizers for each of L considered languages with the same vocabulary size for each of the languages $\frac{N}{L}$. In multilingual tokenization, we apply the tokenizer for a specific language separately and produce a token with language identification.[3] The vocabulary consists of L

[3] Only the special tokens are shared across languages, e.g.,

segments of total size N. Naturally, the tokenized texts in different languages will consist of tokens from distinct vocabulary segments. Noticeably, the same character sequence in different languages can be assigned different token ids.

Language-Mixed Tokenization TOKMIX: We train Unigram LM tokenizers for each of L languages. Subsequently, we averaged vocabulary unit probabilities across tokenizers, sorted them, and trimmed the vocabulary to the pre-set vocabulary size N keeping the units with the highest probability. [4]

$$\hat{\theta} = \sum_{i=1}^{L} w_i \theta_i \tag{1}$$

w_i are weights assigned to each language. By default, we set the weights to be uniform and equal to $\frac{1}{L}$. Unlike NOOVERLAP, the same vocabulary units coming from distinct monolingual tokenizers are merged into one unit with averaged probability.

2.3 Tokenizer and Model Training Setting

We initially focused on a group of 6 languages varying both in the script and language family: Arabic, Chinese, Greek, Turkish, Spanish, and English. In subsequent experiments, we extend the method to 20 languages.

We download 10% of CC corpus available atv https://data.statmt.org/cc-100/. Following the methodology in (Conneau and Lample, 2019), we subsample each language's data to ensure that the training corpus is well-balanced across languages. An equation defines the sample size c_l for language l:

$$c_{l,\alpha} = c_{\min} \cdot \left(\frac{|C_l|}{c_{\min}} \right)^{\alpha} \tag{2}$$

Where $c_{\min}$ is the minimal sample size (defined by the smallest language), and C_l is all data available for a language, α is the so-called "balancing parameter". In our experiments, we set $c_{\min}$ to 10 M characters, C_l is, e.g., 8.8 B characters for English. We set α to 0.25, which corresponds to a balancing factor picked for XLM-Roberta (Conneau et al., 2019). The training data for the tokenizer and the model are the same. The vocabulary size N was set to 120,000. Appendix A contains technical details about our approach.

3 Measuring Tokenizer Properties

This section presents our in-depth analytical approach to evaluate different aspects of multilingual tokenization. We introduce non-parametric measures that describe the key properties of multilingual tokenizers: quality of vocabulary representation for particular languages and lexical overlap across languages.

We base our analysis on the empirical probability distribution of vocabulary units $v \in \mathcal{V}$ computed on training corpus for each language l:

$$d_{l,\mathcal{V}}(v) = \frac{f(v, C_l)}{\sum_{v \in \mathcal{V}} f(v, C_l)} \tag{3}$$

Function $f(v, C_l)$ is the number of occurrences of a vocabulary unit v in monolingual training corpus C_l.

3.1 Vocabulary Allocation

We aim to quantify how well multilingual vocabulary represents meaningful lexical units of particular languages. Our intuition is that a good lexical representation is obtained when: 1. It uses a vast portion of multilingual vocabulary, and thus a larger part of the embedding layer is devoted to the language; 2. The text in the language is split into longer and potentially more meaningful tokens.

Vocabulary Allocation: Average Rank To measure the number of vocabulary units available for modeling specific languages, we propose an estimation of the average rank of vocabulary units in distribution over a monolingual corpus.[5] This measure denotes how many tokens are typically considered by a language model that has access to language identity information but no context (probabilistic unigram LM).

$$\mathrm{AR}_{l,\mathcal{V}} = \sum_{v \in \mathcal{V}} \mathrm{rank}(v, d_{l,\mathcal{V}}) d_{l,\mathcal{V}}(v) \tag{4}$$

Our intuition is that model will have better information about the language's lexicon when vocabulary is distributed over a larger number of tokens as more parameters of the input embedding layer would be allocated to represent language-specific features. Moreover, larger vocabularies tend to cover longer and more meaningful units.

"<s>" – the beginning of a sentence token.

[4] To account for possible overlaps between language-specific vocabularies, we set their sizes above $\frac{N}{L}$. It assures that joint vocabulary will have at least N tokens.

[5] In this context, rank is the position of unit v in the vocabulary $\mathcal{V}$ sorted in descending order by the probability distribution $d_{l,\mathcal{V}}$

Vocabulary Allocation: Characters per Token In line with previous intuition, longer tokens have a more meaningful representation. Therefore, we measure text fragmentation by computing the average number of characters for a vocabulary unit in monolingual corpus C_l.:

$$\text{CPT}_{l,\mathcal{V}} = \frac{|C_l|}{|T_{\mathcal{V}}(C_l)|} \tag{5}$$

$T_{\mathcal{V}}(C_l)$ is the tokenization of the corpus with vocabulary $\mathcal{V}$; $|C_l|$ is the size of the corpus measured as the number of characters. We choose the number of characters as the unit to relate to because it's not susceptible to cross-lingual differences regarding word boundaries and the average length of words. Still, the amount of information conveyed by a single character varies largely with the writing systems, e.g., texts written in logographic scripts (e.g., Chinese, Japanese) tend to be shorter in the number of letters than similarly informative ones in the phonetical script (e.g., Latin) (Perfetti and Liu, 2005).

3.2 Vocabulary Overlap

Another important property of multilingual vocabulary is sharing lexical units across languages. Previous works claimed that vocabulary overlap improves cross-lingual transfer for learning downstream tasks (Pires et al., 2019; Wu and Dredze, 2019). We measure overlap as the divergence between corpora distributions d_l (defined in equation 3). We use the Jensen-Shanon divergence.[6] We apply JSD because it is symmetric and applicable for distribution with different supports. The latter is often the case when distributions are estimated for languages with distinct writing systems.

$$\begin{aligned}\text{JSD}(d_{l1,\mathcal{V}}||d_{l2,\mathcal{V}}) = \\ = \frac{1}{2}\sum_{v \in \mathcal{V}} d_{l1,\mathcal{V}}(v) \log_2 \frac{d_{l1,\mathcal{V}}(v)}{m_{l1,l2,\mathcal{V}}(v)} + \\ + \frac{1}{2}\sum_{v \in \mathcal{V}} d_{l2,\mathcal{V}}(v) \log_2 \frac{d_{l2,\mathcal{V}}(v)}{m_{l1,l2,\mathcal{V}}(v)}\end{aligned} \tag{6}$$

where:

$$m_{l1,l2,\mathcal{V}} = \frac{1}{2}d_{l1,\mathcal{V}} + \frac{1}{2}d_{l2,\mathcal{V}} \tag{7}$$

JSD is bounded in the range 0 to 1. The lower the value, the larger the overlap across corpora.

Another possibility to quantify overlap is to count unique vocabulary units appearing in tokenized texts across languages. The advantage of divergence is that it reflects the frequency of shared tokens across corpora. It is also less affected by the choice of the data size used for estimating empirical probability distributions (d_l).

4 Evaluating Language Modeling and Downstream Tasks

In this section, we present the tasks and measures for evaluation of multilingual language models trained with different tokenizers.

4.1 Language Modeling

We evaluate the masked language modeling performance with mean reciprocal rank:

$$\text{MRR} \quad = \quad \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\text{rank}(x_i, \hat{P}(\cdot|X \setminus x_i))} \tag{8}$$

where $\hat{P}(\cdot|X \setminus x_i)$ is the probability over vocabulary of predicting token x_i by the model given its context: $X \setminus x_i$.

4.2 Downstream Evaluation

The downstream tasks are taken from the XTREME (Hu et al., 2020), which is the collection of diverse datasets with predefined splits used to evaluate multilingual models' representation.

We probe the models' output representation to evaluate how useful the learned representation is for the downstream tasks. Only an additional linear layer is trained for the task, while the base model representation is frozen. The approach is suitable for evaluating how well the pre-trained model encodes linguistic phenomena as it does not change parameters learned in pre-training in contrast to regular fine-tuning (Conneau et al., 2018a; Belinkov, 2022).

Word-level Tasks The first set of tasks covers classification on a single word or word pair level. The probe is a linear layer taking word representations on input and outputting one of the classes. For word representations, we take the model's output embedding of the first subwords. We evaluate the results with an F1 score averaged across classes (macro-average).

[6] In NLP literature, JSD is also known as "information radius" (Manning and Schütze, 2001).

		ar	tr	zh	el	es	en
AR	Unigram	2129	2719	**5919**	2070	1439	1513
	BPE	2972	3226	4294	**2907**	**2220**	**2143**
	NoOverlap	2537	2653	2090	2065	1661	1597
	TokMix	**3485**	**4167**	3961	2639	1999	1898
CPT	Unigram	3.16	4.01	1.84	3.5	3.88	3.91
	BPE	**3.7**	4.19	**2.03**	**3.97**	**4.34**	**4.22**
	NoOverlap	3.53	4.19	1.56	3.81	4.15	4.15
	TokMix	**3.7**	**4.45**	1.73	3.9	4.24	4.18

Table 1: Values of *vocabulary allocation* measures for 4 tokenizers trained on the small language set. The highest values for each language are bolded.

We test syntactic tasks: **Part of Speech** and **Dependency labeling** on Universal Dependencies (de Marneffe et al., 2021) and **Named Entity Recognition** on Wikiann dataset (Pan et al., 2017). In dependency labeling, we use edge probe (Tenney et al., 2019) on top of the representation of two words connected by the dependency arc.

Sentence-level Tasks In this set of tasks, we examine whether the model learns sentence-level representations that capture its semantics and can be transferred across languages. To obtain this sentence embedding, we average the model's output representation across all the tokens in the sentence.

We evaluate **Natural Language Inference** on XNLI dataset (Conneau et al., 2018b) and **Sentence Retrieval** on Tatoeba bitext corpus (Artetxe and Schwenk, 2019). For NLI, we use edge probing. Sentence retrieval is solved by an unsupervised algorithm matching sentences based on their cosine similarity. In Appendix A.3, we provide details of the datasets and probe training.

4.2.1 In-language vs. Cross-lingual Transfer

For all the downstream tasks, except sentence retrieval, we compute in-language performance by training the probe and evaluating it on held-out test data in the same language. We quantify cross-lingual transfer by training a probe on one language (source) and evaluating it on the test set for another language (target).

5 Experiments and Results

We train four tokenizers for the smaller set of diverse 6 languages (en, es, tr, el, zh, ar) using existing methods: Unigram, BPE, and our methods for monolingual tokenizer merging: NoOverlap, TokMix. Using these tokenizers, we then train four models[7] following the settings of XLM-Roberta (Conneau et al., 2019) which we then use for the probing experiments.

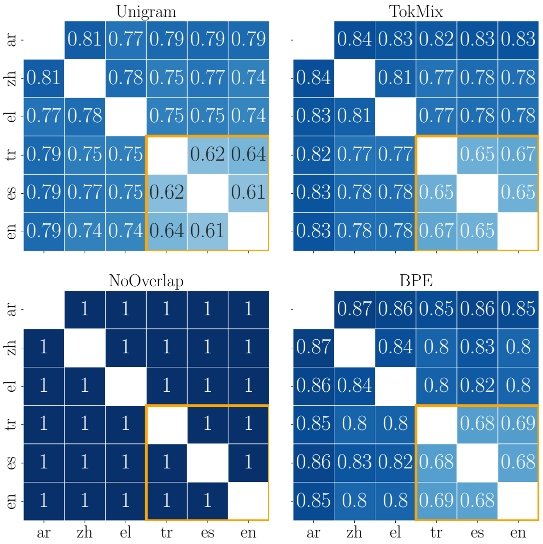


Figure 2: *Vocabulary overlap* measure: Jensen-Shanon divergence for four tokenization methods. Orange square in the bottom right groups the languages with the same script (Latin).

In Section 5.1, we analyze the distribution of learned vocabulary units and compute *vocabulary allocation* and *vocabulary overlap* measures described in Section 3. Then in Section 5.2, we evaluate the models' performance measures introduced in Section 4 and compare them with the measures for tokenizers.

Subsequently, we repeat the analysis for the broader set of 20 diverse languages (including six mentioned earlier and: he, ka, ur, hi, mr, th, ta, te, bg, ru, sw, vi, fr, de) with three tokenization methods used in three pre-trained models. In this setting, we do not use NoOverlap tokenizer, which cannot be trained effectively due to the necessity of constraining vocabulary for each language to $\frac{N}{L} = 6,000$.

5.1 Evaluation of Tokenizers' Properties

***Vocabulary allocation* largely varies throughout languages and tokenization methods.** Table 1 shows that the average rank noticeably differs across languages. The highest AR is observed for Chinese, which is caused by the fact that logographic scripts require an extensive vocabulary capacity to encode all characters.

Multilingual *vocabulary allocation* is highly dependent on the tokenization method used. Vocabulary learned with Unigram underperforms BPE and

[7] Details about the pretraining and probing procedures are described in Appendix A.2

	V. Allocation		MLM	NER	POS	Dep. labeling	NLI
	(AR)	(CPT)	(MRR)	(F1)	(F1)	(F1)	(Acc)
Unigram	2042	3.17	42.0	62.8 ±0.1	57.1 ±0.2	48.1 ±0.4	**53.4** ±0.5
BPE	2193	**4.47**	35.6	**70.4** ±0.1	**68.9** ±0.2	**58.7** ±0.4	**53.3** ±0.3
NoOverlap	1829	3.16	**42.7**	69.4 ±0.1	**69.2** ±0.2	**58.8** ±0.3	**53.0** ±0.4
TokMix	**2198**	3.34	38.7	**70.2** ±0.1	67.3 ±0.1	57.3 ±0.4	**53.3** ±0.4

(a) 6 languages

	V. Allocation		MLM	NER	POS	Dep. labeling	NLI
	(AR)	(CPT)	(MRR)	(F1)	(F1)	(F1)	(Acc)
Unigram	623	2.89	**52.6**	58.9 ±0.2	54.0 ±0.4	43.7 ±0.4	**53.2** ±0.3
BPE	**809**	**3.43**	40.5	**66.3** ±0.2	**67.3** ±0.4	**54.5** ±0.5	**53.5** ±0.3
TokMix	689	3.23	44.8	65.4 ±0.3	**66.5** ±0.4	**53.9** ±0.5	**52.3** ±0.3

(b) 20 languages

Table 2: Averaged results of evaluation for in-language properties and tasks. Each probing result is an average of 5 random seeds (for 6 languages) and 3 random seeds (for 20 languages). The best value in each metric is underlined, and bolded results are closer than the sum of standard deviations from the optimal value.

	V. Allocation		MLM
	(AR)	(CPT)	(MRR)
CPT	**0.790**	-	-
MRR	**-0.723**	**-0.913**	-
NER	**0.394**	**0.657**	**-0.745**
POS	0.320	**0.724**	**-0.754**
Dep l.	0.266	**0.675**	**-0.695**
NLI	**0.56**	0.388	**-0.437**

Table 3: Spearman correlations between task coefficients for in-language results and tokenizer measures. Statistically significant correlations ($p < 0.01$) are bolded. Computed for 20 languages.

TOKMIX in both average rank and character per token. Table 7 presented in the Appendix shows that this trend exists throughout languages except for Chinese. This suggests that our vanilla Unigram is a suboptimal multilingual vocabulary learner.

It is important to note that NOOVERLAP scores even lower than Unigram in the *vocabulary allocation* measures due to the limited vocabulary size for each language and disallowing overlap. However, as shown in the next sections, LM trained with this tokenizer can achieve good results on some tasks.

The choice of tokenization method affects *vocabulary overlap*. Figure 2 shows Jensen-Shanon divergencies between the vocabularies of six languages. We observe that the highest cross-lingual overlaps appear in the vocabulary obtained by Unigram, followed by TOKMIX, and BPE. Expectedly, we do not observe overlaps for NOOVERLAP's setting (JSD $= 1$).

Jensen-Shanon divergence is a good predictor of whether the languages share the script. For all tokenization methods, the divergence is significantly smaller in the bottom-right square grouping of the languages using Latin script. This effect is even more visible in the visualization of JSD computed for twenty languages (Figure 8 in Appendix C).

5.2 Tokenizer Properties Impact Language Model's Performance

High *vocabulary allocation* improves downstream results for word-level tasks. In Table 2a, we observe that the choice of the tokenization method significantly impacts the results for POS, dependency labeling, and NER. We presume it results from learning good lexical representations throughout languages, e.g., by BPE and TOKMIX. The higher *vocabulary allocation* is especially beneficial for word-level tasks. Whereas the influence on the sentence-level task (NLI) is minimal.

Notably, the model instance with NOOVERLAP tokenizer achieves the best F1 in POS and dependency labeling despite underperforming in *vocabulary allocation*. It is the result of learning language-specific representation for tokens that is especially useful for syntactic tasks.

Better MLM performance doesn't bring improvement to downstream tasks. In Table 2a, we observe that the models performing better on masked token prediction (MRR) tend to be worse on downstream tasks (POS and NER). It is the result of different average ranks. The higher it is, the more vocabulary units a language model needs to consider for masked token filling, making

Metric	Tokenizer	Different script	Same script	All transfers
Overlap (JSD)	Unigram	**0.77**	**0.62**	**0.74**
	BPE	0.83	0.68	0.8
	NoOverlap	1.0	1.0	1.0
	TokMix	0.8	0.65	0.77
NER (F1)	Unigram	31.3 ±0.4	55.4 ±0.2	36.1 ±0.4
	BPE	**33.5** ±0.5	**59.9**±0.2	**38.7** ±0.4
	NoOverlap	32.0 ±0.5	48.6 ±0.4	35.3 ±0.5
	TokMix	31.8 ±0.4	58.0 ±0.3	37.0 ±0.4
POS (F1)	Unigram	18.1 ±0.4	38.3 ±0.4	22.2 ±0.4
	BPE	**25.8** ±0.5	**40.8** ±0.4	**28.8** ±0.5
	NoOverlap	20.1 ±0.5	**41.9** ±0.5	24.5 ±0.5
	TokMix	21.9 ±0.4	40.4 ±0.3	25.6 ±0.4
Dep. labeling (F1)	Unigram	11.1 ±0.3	25.5 ±0.3	14.0 ±0.3
	BPE	**15.9** ±0.4	**27.0** ±0.4	**18.1** ±0.4
	NoOverlap	12.8 ±0.4	**27.8** ±0.5	15.8 ±0.4
	TokMix	12.6 ±0.5	26.1 ±0.3	15.3 ±0.5
NLI (Acc)	Unigram	**42.2** ±0.7	**43.7** ±0.7	**42.5** ±0.7
	BPE	**42.4** ±0.7	**45.2** ±0.8	**43.0**±0.7
	NoOverlap	37.3 ±0.6	37.1 ±0.5	37.2 ±0.6
	TokMix	**41.2** ±0.7	**42.7** ±0.5	**41.5** ±0.7
Retrieval (Acc)	Unigram	21.0	**43.9**	25.6
	BPE	20.9	40.7	24.9
	NoOverlap	12.3	28.0	15.4
	TokMix	**23.0**	43.4	**27.1**

(a) 6 languages

Tokenizer	Different script	Same script	All transf
Unigram	**0.75**	**0.58**	**0.73**
BPE	0.83	0.67	0.81
TokMix	0.8	0.64	0.78
Unigram	33.2 ±0.5	50.7 ±0.6	35.4 ±0.5
BPE	**36.6** ±0.6	**54.3** ±0.3	**38.8** ±0.5
TokMix	**36.5** ±0.6	**53.7** ±0.5	**38.7** ±0.6
Unigram	23.4 ±0.5	32.9 ±0.3	24.6 ±0.5
BPE	**30.5** ±0.6	**40.7** ±0.4	**31.8** ±0.6
TokMix	**29.2** ±0.5	**40.4** ±0.3	**30.7** ±0.5
Unigram	13.0 ±0.6	15.6 ±0.5	13.4 ±0.6
BPE	**16.5** ±0.6	**19.2** ±0.5	**16.9** ±0.5
TokMix	**16.0** ±0.5	**19.4** ±0.4	**16.5** ±0.5
Unigram	**37.3** ±0.5	37.5 ±0.4	37.4 ±0.5
BPE	36.2 ±0.5	**38.7** ±0.5	**36.7** ±0.5
TokMix	**37.8** ±0.5	**39.2** ±0.5	**38.1** ±0.5
Unigram	**44.1**	44.4	44.2
BPE	**44.1**	**49.1**	**45.1**
TokMix	42.8	46.9	43.6

(b) 20 languages

Table 4: Averaged results of the evaluation for cross-language overlaps and transfers. Each probing result is an average of 5 random seeds (for 6 languages) and 3 random seeds (for 20 languages). The best value in each metric is underlined, and bolded results are closer than the sum of standard deviations from the optimal value.

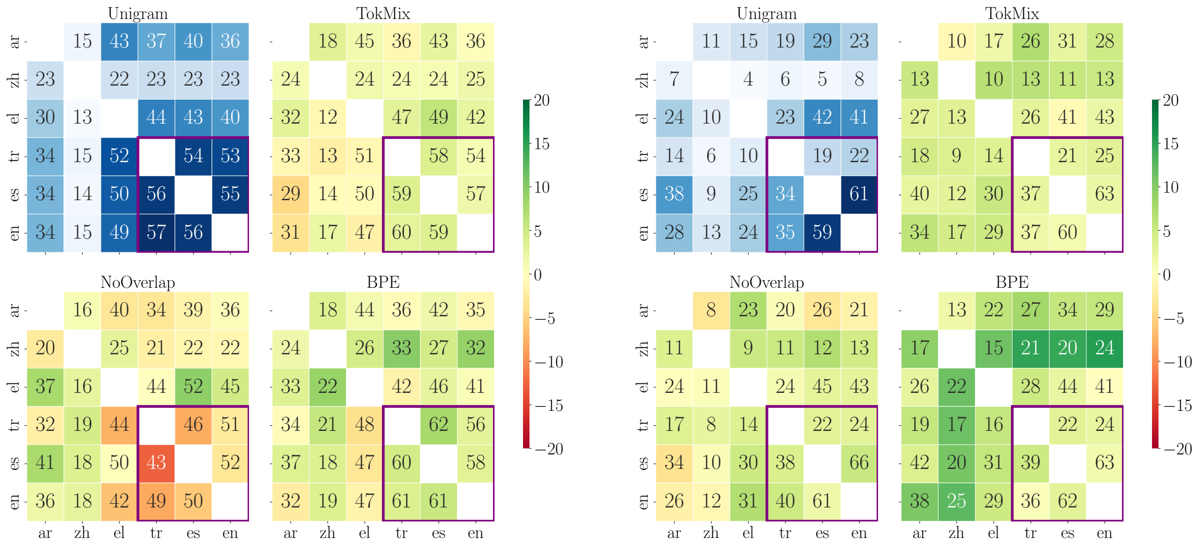
masked word prediction harder. At the same time, a high average rank means that the vocabulary is broader and contains lexical units important for downstream tasks.

Again, this trend does not hold for the results for NOOVERLAP setting, in which the search space for the masked-word problem is limited to the language-specific tokens leading to the best performance in MLM and syntactic tasks (POS and dependency label prediction).

In Table 3, we show that the strong relationship between *vocabulary allocation* (avg. rank and CPT) and LM performance (MRR) is statistically supported. The length of token units has a strong positive influence on POS, dependency labeling, and NER results ($r > 0.65$) and a negative influence on MRR ($r < -0.9$), while it does not significantly affect NLI results. The correlation between the average rank and MRR, NER scores is weaker but still significant. Moreover, it is significantly correlated with XNLI accuracy with a medium coefficient $r = 0.56$, even though the changes in XNLI are low across tokenizers.

Impact of *vocabulary overlap* on cross-lingual transfer varies across tasks. We observed that NOOVERLAP approach obtains competitive results for POS tagging . Surprisingly no vocabulary sharing also improves cross-lingual transfer in the task among languages with Latin script (shown in Table 4a and Figure 3b). We think that the reason behind the strength of NOOVERLAP approach is that some tokens have different meanings across languages, e.g., the word “a” is an indefinite article in English and a preposition in Spanish.

Nevertheless, vocabulary overlap is crucial to cross-lingual transfer in some tasks. Especially NER within the same script languages (Figure 3a) and sentence-level tasks. For these tasks, NOOVERLAP significantly underperforms other tokenization methods. The drop within Latin script languages is in the range: 6.8 - 11.3% for NER and 12.7 - 15.9% for sentence retrieval. In these cases, usage of the same tokens can indicate that texts refer to the same entities across languages, e.g., names are usually the same strings in the languages sharing writing system.



(a) NER (F1)

(b) POS (F1)

Figure 3: Cross-lingual transfer for POS and NER tasks. The absolute values are presented for the Unigram tokenizer. For other tokenization methods, the color scheme shows a difference from the Unigram algorithm. In the case of NER, we observe a drop in cross-lingual transfer for NoOverlap tokenization, especially for the same script pairs, suggesting that lexical overlap is an important aspect contributing to cross-lingual transfer for NER. We don't see similar drop in the case of Part of Speech tagging.

	V. Overlap	**V. Allocation SRC**		**V. Allocation TGT**	
	(JSD)	(AR)	(CPT)	(AR)	(CPT)
NER	-0.111	**0.249**	**0.33**	0.209	**0.28**
POS	**0.395**	**0.365**	**0.547**	**0.489**	**0.653**
Dep l.	**0.463**	0.19	**0.425**	**0.249**	**0.44**
NLI	**-0.516**	**0.421**	0.203	**0.297**	0.103
Retrieval	**-0.648**	**0.235**	0.082	**0.238**	0.085

Table 5: Spearman correlations between cross-lingual transfer results and tokenization measures. *vocabulary overlap* is measured by JSD, we also measure the correlation with *vocabulary allocation*s of source and target language of the transfer directions. Statistically significant correlations ($p < 0.01$) are bolded. Computed for six languages.

Table 5 presents the correlations for cross-lingual transfer scores with JSD measuring *vocabulary overlap*. The coefficient supports our previous observation that lower overlap (thus higher JSD) improves transfer for POS tagging and dependency labeling and deteriorates it for other tasks. Although, the correlation for NER is not significant. The *vocabulary allocation*s of source and target languages significantly influence the cross-lingual transfers. Similarly to the in-language correlations, the influence of character per token is more substantial on word-level tasks, while Average Rank affects sentence-level tasks to a larger extent. This observation underlines the importance of allocating a sufficient portion of vocabulary for low-resource for better cross-lingual transfer. [8]

Results generalize to the larger set of languages. The key observation for six language sets holds in the model trained for twenty languages. Table 2b shows that BPE and TokMix obtain better *vocabulary allocation* than Unigram leading to improved results for word-level downstream tasks (NER, POS, Dependency labeling). Due to the smaller vocab size to the language number ratio, average ranks decrease for all methods.

We observe in Table 4b that the cross-language

[8] We describe the correlation analysis in detail in Appendix C.3.

vocabulary overlap is the highest for Unigram and lowest for BPE, similar to the six languages settings. However, the association between *vocabulary overlap* and the cross-lingual transfers is less pronounced.

6 Related Work

Importance of *vocabulary overlap*. Wu and Dredze (2019); Pires et al. (2019) claimed that multilingual overlap benefits cross-lingual transfer. In contrast to this work, they compare overlaps for different language pairs with only one tokenizer. We think that their observations may be confounded by the typological similarity between languages. In the following works, Conneau et al. (2020) found that sharing parameters in top layers is more important to multilingualism than same token embedding. Similar results were demonstrated by Wang et al. (2021); Dufter and Schütze (2020) who show that in bilingual models, artificially removing *vocabulary overlap* (similarly to ours NoOverlap) does not deteriorate cross-lingual transfer. In contrast to many previous approaches, we used probing for evaluation because this method offers better insight into representation learned in pre-training. Similarly, our results, Malkin et al. (2022); Limisiewicz et al. (2022) observed that differences in scripts could, in some cases, improve the cross-lingual transfer in masked language modeling and for downstream tasks.

Importance of *vocabulary allocation*. The effect of *vocabulary allocation* on model performance was studied to a lower extent. Zheng et al. (2021) observed that limited vocabulary capacity allocated for specific languages impedes the downstream tasks' performance and thus proposed a method to obtain more balanced *vocabulary allocation* throughout languages. For the same purpose, Chung et al. (2020) proposed a novel approach to generating multilingual vocabulary based on clustering the target languages and merging separate vocabularies. Recently, Liang et al. (2023) based on the elements of both approaches and increased vocabulary to train the XLM-V model, achieving better results than its predecessor (XLM-Roberta Conneau et al. (2019)).

In a monolingual setting, Bostrom and Durrett (2020) argued that Unigram tokenization produces subword tokens that are more aligned with morphological units that bring improvement for downstream tasks. This contrasts with our finding of Unigram's underperformance when applied to a multilingual corpus.

Improving multilingual sub-word tokenization. Patil et al. (2022) proposed a modification to BPE algorithm that increases overlap between similar languages and benefits cross-lingual transfer. Rust et al. (2021) observed that models with dedicated monolingual tokenizers outperform multilingual ones. This observation can be utilized by adapting the embedding layer of the model for a target language (Pfeiffer et al., 2020; Artetxe et al., 2020; Minixhofer et al., 2022). However, these approaches require language-specific modification of the model, limiting its multilingual aspect.

Alternatives to sub-word tokenization. There are multiple alternative approaches for inputting text into deep models, such as character-based representation (Clark et al., 2022), byte input (Xue et al., 2022), or representing the input text as images (Salesky et al., 2021). Mielke et al. (2021) summarize a wide range of methods and point out that they offer trade-offs and may be better suited for certain tasks or languages.

7 Conclusions

We introduced a new framework for the evaluation of multilingual subword tokenizers. We show that *vocabulary allocation* is a crucial aspect affecting the results of many downstream tasks. Specifically, we have observed the following trends: 1. Including longer and more diverse vocabulary units (higher *vocabulary allocation*) improves in-language results and cross-lingual transfers for word-level tasks; 2. *vocabulary overlap* is beneficial for cross-lingual transfer in sentence-level tasks; 3. Among languages with the same script, *vocabulary overlap* improves transfer for NER and deteriorates it for POS and dependency labeling. Our conclusions are in line with the observation of Mielke et al. (2021) that there is no "silver bullet solution" tokenizer suiting all purposes.

We release the code for measuring tokenizer properties: `github.com/tomlimi/entangled_in_scripts`. We believe that it will be a useful evaluation tool for the developers of models who can get a better insight into the tokenization method before computationally expensive model training.

Limitations

To achieve robust, unbiased results, we decided to train first on a smaller number of languages, fix our methodology and then confirm our findings on the full set of languages. This meant that two rounds of pretraining needed to be done and because of that, we scaled our models down for computational efficiency reasons.

Another limitation of our methodology is the choice to train linear probes on top of the contextualized word representations instead of the more common finetuning approach. Nevertheless, we think that probing gives better insight into the pretrained model's representation.

Ethics Statement

We do not identify ethical risks connected to this work.

Acknowledgements

We thank Jindřich Libovický, Martin Popel, Gabriel Stanovsky, and anonymous ACL reviewers for their valuable comments and suggestions for improvement. This work has been supported by grant 338521 of the Charles University Grant Agency. We have been using language resources and tools developed, stored, and distributed by the LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101).

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the Cross-lingual Transferability of Monolingual Representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637. ArXiv:1910.11856 [cs].

Mikel Artetxe and Holger Schwenk. 2019. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610. ArXiv:1812.10464 [cs].

Yonatan Belinkov. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Comput. Linguistics*, 48(1):207–219.

Kaj Bostrom and Greg Durrett. 2020. Byte Pair Encoding is Suboptimal for Language Model Pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4617–4624. Association for Computational Linguistics.

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. *Trans. Assoc. Comput. 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. 2019. Unsupervised Cross-lingual Representation Learning at Scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018a. What You Can Cram Into a Single $&!#* Vector: Probing Sentence Embeddings for Linguistic Properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018b. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2475–2485. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging Cross-lingual Structure in Pretrained Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6022–6034. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. *Comput. Linguistics*, 47(2):255–308.

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.

Philipp Dufter and Hinrich Schütze. 2020. Identifying Elements Essential for BERT's Multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. *CoRR*, abs/2003.11080.

Taku Kudo. 2018. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. 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.

H. W. Kuhn. 1955. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

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. *CoRR*, abs/2301.10472.

Tomasz Limisiewicz, Dan Malkin, and Gabriel Stanovsky. 2022. You Can Have Your Data and Balance It Too: Towards Balanced and Efficient Multilingual Models. *CoRR*, abs/2210.07135.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4903–4915. Association for Computational Linguistics.

Christopher D. Manning and Hinrich Schütze. 2001. *Foundations of Statistical Natural Language Processing*. MIT Press.

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. *ArXiv*, abs/2112.10508.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective Initialization of Subword Embeddings for Cross-Lingual Transfer of Monolingual Language Models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3992–4006. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual Name Tagging and Linking for 282 Languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1946–1958. Association for Computational Linguistics.

Vaidehi Patil, Partha P. Talukdar, and Sunita Sarawagi. 2022. Overlap-based Vocabulary Generation Improves Cross-lingual Transfer Among Related Languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 219–233. Association for Computational Linguistics.

Charles A. Perfetti and Ying Liu. 2005. Orthography to Phonology and Meaning: Comparisons Across and Within Writing Systems. *Reading and Writing*, 18(3):193–210.

Jonas Pfeiffer, Ivan Vulic, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7654–7673. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How Multilingual is Multilingual BERT? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4996–5001. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively Multilingual Transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulic, Sebastian Ruder, and Iryna Gurevych. 2021. How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 3118–3135. Association for Computational Linguistics.

Elizabeth Salesky, David Etter, and Matt Post. 2021. Robust Open-Vocabulary Translation from Visual Text Representations. 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 7235–7252. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast WordPiece Tokenization. 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 2089–2103. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What Do You Learn From Context? Probing for Sentence Structure In Contextualized Word Representations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 473–482. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, pages 38–45. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 833–844. 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. ArXiv:2105.13626 [cs].

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.

A Technical Details

A.1 Tokenizer training details

We use the Huggingface Tokenizers library for training the Unigram and BPE tokenizers. We kept the default values for the training parameters. Namely, for Unigram, we use a maximum piece length of 16 and a shrinking factor of 0.75. For BPE, we use alphabet size 1000 and minimum merge frequency 2. For all languages, we use SentencePiece (Kudo and Richardson, 2018) for word segmentation techniques instead of language-specific word tokenizers.

A.2 Model Architecture and Pre-Training

In this study, we employed the Huggingface library (Wolf et al., 2020) to conduct all experiments. The model architecture is based on XLM-Roberta, although for our purposes, it was scaled down. Specifically, the size of the embeddings is 768, the number of attention layers is 8, and the number of attention heads is 6. The maximum sentence length is 128, and the vocabulary size is 120000. The number of parameters is 150M and, therefore, roughly 2 times smaller than the XLM-Roberta base model.

The model was pre-trained for 10 epochs with a batch size of 1024. The learning rate was 5e-5 with linear decay and weight decay and 1% warm-up steps. In pretraining, we used AdamW optimizer (Loshchilov and Hutter, 2019).

In total, we pretrained 7 models. The models were trained on 3 Nvidia GPUs. The probing experiments were run on 1 Nvidia GPU with 40GB of memory (Nvidia A40). The pretraining took about 17 hours for each 6-language model and 60 hours for the models trained on the full set of 20 languages.

We didn't pursue any extensive hyperparameter search efforts as this was not the focus of our work. We selected the best batch size and learning rates for the pre-training based on a few trials.

A.3 Downstream Data and Training

The probes were for 30 epochs with early stopping and batch size 16. We used an initial learning rate of 2e-5. Other training parameters were the same as in pretraining. Probing experiments took between 5 to 180 minutes to complete on the same infrastructure as used for pretraining. We ran around 360 probe trainings.

POS We use Part of Speech annotations from Universal Dependencies (de Marneffe et al., 2021). The dataset is available for 17 languages analyzed by us (not covered: Swahili, Thai, Georgian). Each word is assigned one of the 17 coarse POS tags.

NER We use Wikiann dataset (Pan et al., 2017) consisting of Wikipedias article with annotated named entities of three types: location, person, and organization in IOB2. Following XTREME, we use balanced data splits from (Rahimi et al., 2019).

Dependency labeling As in Part of Speech, we use Universal Dependencies (de Marneffe et al., 2021) for the dependency relation annotations. We use the largest UD treebank available for each language. For each word we predict one of the 37 universal relations to its head word. Because the relation is between two words, we use the concatenation of the two word representations along with their element-wise product as an input to the probe ($[h_{w1}; h_{w2}; h_{w1} \odot h_{w2}]$).

NLI We use XNLI dataset (Conneau et al., 2018b) for Natural Language Inference. We train the linear classification probe on top of the concatenation of two sentence vectors and their element-wise product: $[h_{s1}; h_{s2}; h_{s1} \odot h_{s2}]$. We predict one of two relations between the first of sentences (called premise): contradicts, entails, or is neutral to the second sentence (called a hypothesis). We evaluate XNLI with the accuracy of classification.

XNLI contains data for 15 languages (not covered: te, ta, mr, he, ka).

Sentence Retrieval We use up to 1,000 sentences aligned for pairs of languages from Tatoeba dataset (Artetxe and Schwenk, 2019). For the pairs including English, we use the same sample as in XTREME data collection. For other pairs, we perform sampling ourselves.

We compute the cosine similarity between sentence representations across languages and find the best alignment with the Hungarian algorithm(Kuhn, 1955). We compute the accuracy as the number of correctly aligned sentences divided by the total number of sentences.

B In-depth Tokenizers Analysis

In Figure 4, we present the probabilities of vocabulary units, computed on concatenate six languages corpora, learned by different tokenization

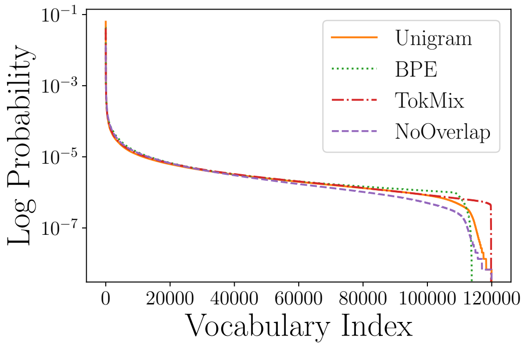


Figure 4: Log-probabilites of vocabulary units in decreasing order for four tokenization methods.

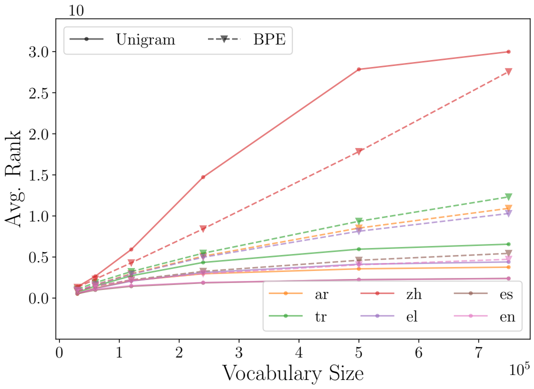


Figure 5: Avearage Rank measured for vocabularies of different sizes, obtained with BPE and Unigram algorithms.

algorithms. Unigram and NoOverlap use a bigger fraction of the vocabulary for rarely appearing tokens (with probability lower than 10^{-6}). BPE and TokMix produce a vast set of tokens with probabilities in the range between 10^{-5} and 10^{-6}. Interestingly, the former algorithm allocates about 6000 vocabulary entries to tokens not appearing in the corpora.

BPE is better than Unigram in *vocabulary allocation* throughout languages. To support this claim, we train Unigram and BPE tokenizers for different vocabulary sizes. We observe that both the average rank (Figure 5) and CPT (Figure 6) stop rising for vocab sizes above 250,000 (except for Chinese). For BPE, the metrics still steadily rise after this threshold, which makes it overperform Unigram for most languages.

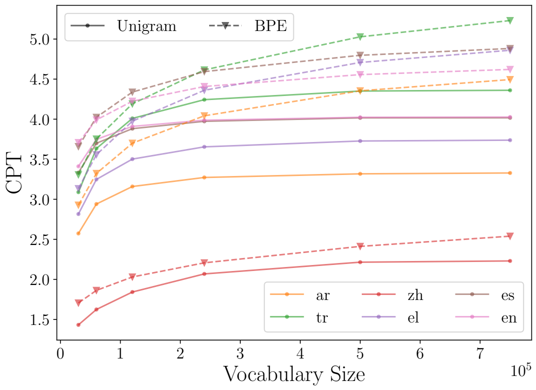


Figure 6: Characters per Token measured for vocabularies of different sizes, obtained with BPE and Unigram algorithms.

	English	Turkish	Greek
Unigram	s, ing, ed, ly, d, If	n, a, e, k, s, i	η, ς, ο, α, ή, ει
BPE	▁the, ▁to, ▁of, ▁and, ▁If,▁a	▁o, ▁veyaim, im inin, ası, esi	▁η, ▁ο, ▁και, ▁ή, ▁να, ▁στον

Table 6: List of units from Unigram and BPE vocabulary with the highest difference in frequency between tokenizers. The first row shows the tokens that appear more frequently in the corpus tokenized by Unigram and the second by the BPE tokenizer. We excluded punctuation marks and special characters from the list.

We think that the reason why Unigram does not learn valuable tokens after this point is the way the initial vocabulary is constructed, i.e., it is the set of all character n-grams appearing in the corpus with n lower than 16. In contrast to BPR, Unigram's vocabulary won't cover longer words than 16 characters, which are useful in modeling some languages.

We believe that further work on identifying optimal strategies for multilingual tokenization is needed.

Vocabulary units preferred by tokenizers. In Table 6, we show the tokens with the highest differences in empirical probabilities obtained with BPE and Unigram tokenizers for three languages. We see that Unigram prefers suffixes to prefixes. Also, it splits text more often into single, possibly due to lower *vocabulary allocation*.

C Supplementary Results

C.1 Visualizations

We present the additional visualization for the results for transfers across six languages for the tasks not presented in the main text: Dependency labeling 7a and NLI cross-lingual accuracy 7b, Sentence retrieval accuracy 7c.

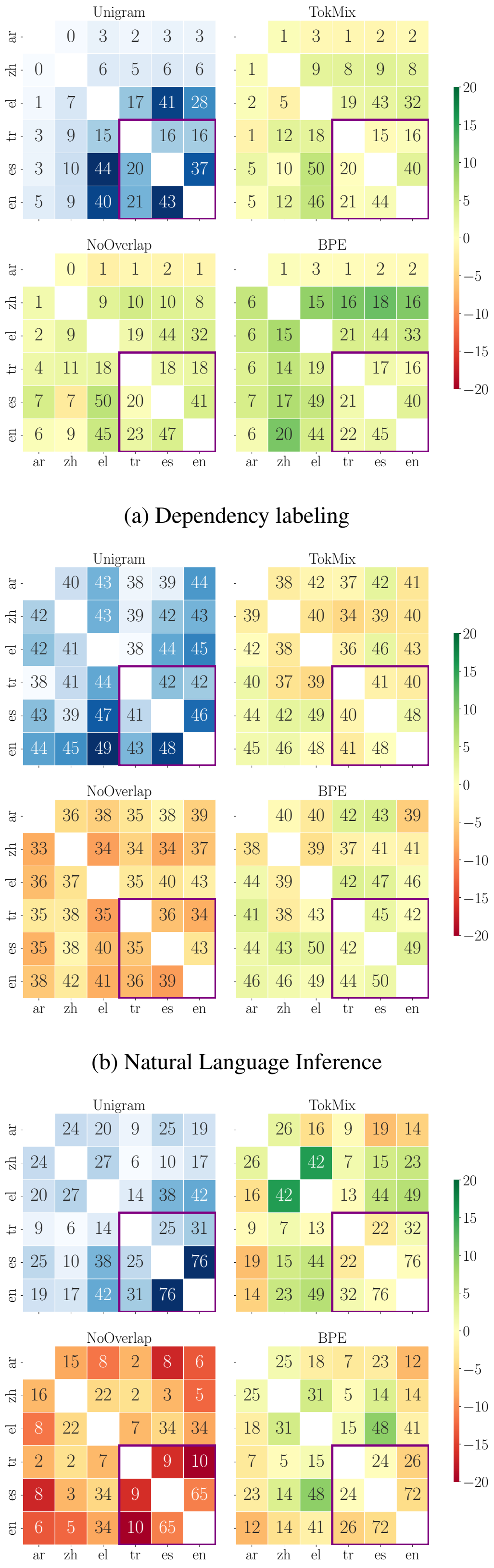
The results of experiments for 20 languages: Jensen-Shanon Divergences 8, and cross-lingual transfers for POS 10a, NER 10b, dependency tree labeling 10c, XNLI 9a, sentence alignment 9b.

C.2 Results for All Languages

We also include detailed results for the in-language experiments along with the proposed tokenizer metrics. In Table 7, we present the results for the six languages.

C.3 Correlation Analysis

We present paired correlation plots for in-language metrics in Figure 11. We use the results from 20 language settings to increase the number of observations. In this analysis, we focus on the differences between the tokenization methods and want to marginalize the language-specific features (such as the pre-training and fine-tuning data size or the model's preference for Indo-European languages). Therefore, for *vocabulary allocation* measures (AR, CPT) and downstream tasks, we subtract the mean for each language. For *vocabulary overlap* measure (JSD) and transfer values, we subtract the mean value for each pair of languages. In both cases, means are computed across all tokenizers. We present Spearman's correlation coefficient and associated p-value.



(a) Dependency labeling

(b) Natural Language Inference

(c) Sentence retrieval

Figure 7: The rest of the 6-language cross-lingual transfer results. The absolute values are presented for the Unigram tokenizer. For other tokenization methods, we show the difference from the unigram algorithm.

		ar	tr	zh	el	es	en	All
metric	tokenizer							
V. Allocation (AR)	Unigram	2129	2719	5919	2070	1439	1513	2042
	BPE	2972	3226	4294	2907	2220	2143	2193
	NoOverlap	2537	2653	2090	2065	1661	1597	1829
	TokMix	3485	4167	3961	2639	1999	1898	2198
V. Allocation (CPT)	Unigram	3.16	4.01	1.84	3.5	3.88	3.91	3.17
	BPE	3.7	4.19	2.03	3.97	4.34	4.22	4.47
	NoOverlap	3.53	4.19	1.56	3.81	4.15	4.15	3.16
	TokMix	3.7	4.45	1.73	3.9	4.24	4.18	3.34
MLM (MRR)	Unigram	36.0	36.0	34.2	46.3	49.7	49.6	42.0
	BPE	28.7	33.6	28.6	38.6	43.1	41.0	35.6
	NoOverlap	38.1	39.6	41.4	42.8	47.5	46.6	42.7
	TokMix	31.5	30.6	38.2	41.2	45.3	45.6	38.7
NER (F1)	Unigram	66.4 ±0.1	73.0 ±0.1	35.1 ±0.1	68.0 ±0.1	68.0 ±0.1	66.1 ±0.2	62.8 ±0.1
	BPE	76.1 ±0.0	76.7 ±0.0	54.2 ±0.1	70.3 ±0.1	75.2 ±0.1	70.0 ±0.0	70.4 ±0.1
	NoOverlap	76.5 ±0.1	72.8 ±0.0	58.4 ±0.1	69.6 ±0.1	71.6 ±0.1	67.3 ±0.1	69.4 ±0.1
	TokMix	76.6 ±0.1	76.2 ±0.1	56.1 ±0.0	70.1 ±0.1	74.3 ±0.1	68.1 ±0.1	70.2 ±0.1
POS (F1)	Unigram	54.8 ±0.1	46.9 ±0.2	29.3 ±0.1	52.9 ±0.3	76.5 ±0.2	81.9 ±0.1	57.1 ±0.2
	BPE	66.7 ±0.1	52.1 ±0.1	62.2 ±0.0	63.4 ±0.1	81.7 ±0.4	87.4 ±0.1	68.9 ±0.2
	NoOverlap	66.5 ±0.1	52.5 ±0.2	60.6 ±0.1	67.5 ±0.1	81.3 ±0.6	86.7 ±0.1	69.2 ±0.2
	TokMix	66.0 ±0.1	52.1 ±0.2	56.2 ±0.0	61.7 ±0.2	81.3 ±0.2	86.3 ±0.1	67.3 ±0.1
Dep. labeling (F1)	Unigram	13.5 ±0.6	58.6 ±0.8	20.7 ±0.1	58.4 ±0.4	71.9 ±0.1	65.7 ±0.2	48.1 ±0.4
	BPE	13.8 ±0.0	63.7 ±1.2	59.5 ±0.1	68.2 ±0.8	77.0 ±0.2	70.3 ±0.4	58.7 ±0.4
	NoOverlap	13.2 ±0.0	65.0 ±0.5	60.5 ±0.2	67.7 ±0.2	77.1 ±0.3	69.2 ±0.3	58.8 ±0.3
	TokMix	14.1 ±0.0	62.9 ±1.2	53.8 ±0.1	67.3 ±0.5	76.5 ±0.1	69.1 ±0.2	57.3 ±0.4
NLI (Acc)	Unigram	52.5 ±0.3	52.9 ±0.3	47.5 ±1.4	55.0 ±0.2	55.3 ±0.3	57.4 ±0.5	53.4 ±0.5
	BPE	52.2 ±0.3	53.6 ±0.5	45.2 ±0.4	55.6 ±0.3	55.7 ±0.2	57.8 ±0.2	53.3 ±0.3
	NoOverlap	52.9 ±0.7	54.0 ±0.2	44.0 ±0.8	54.8 ±0.1	54.9 ±0.3	57.3 ±0.3	53.0 ±0.4
	TokMix	52.0 ±0.2	53.6 ±0.5	46.2 ±1.0	55.4 ±0.3	55.3 ±0.1	57.5 ±0.2	53.3 ±0.4

Table 7: Results of evaluation for in-language properties and tasks for six diverse languages. We observe significant changes for different tokenization methods. The results for MRR, POS, NER, XNLI are in percent. For the downstream task, we show average and standard deviations computed for five runs of probing.

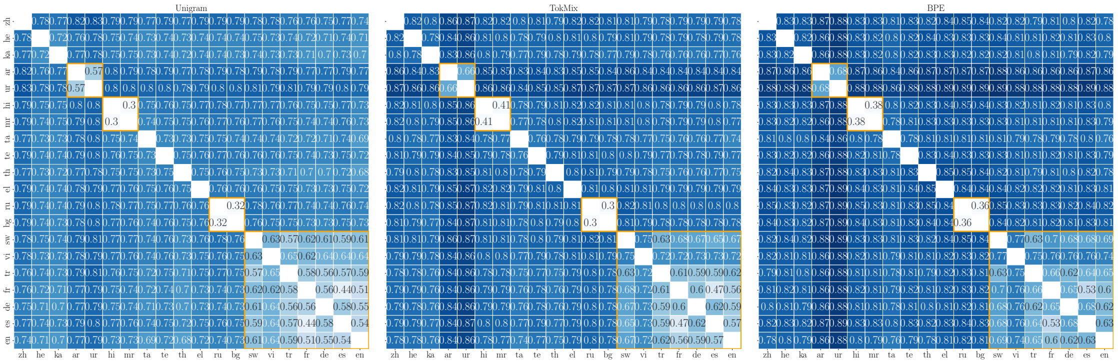
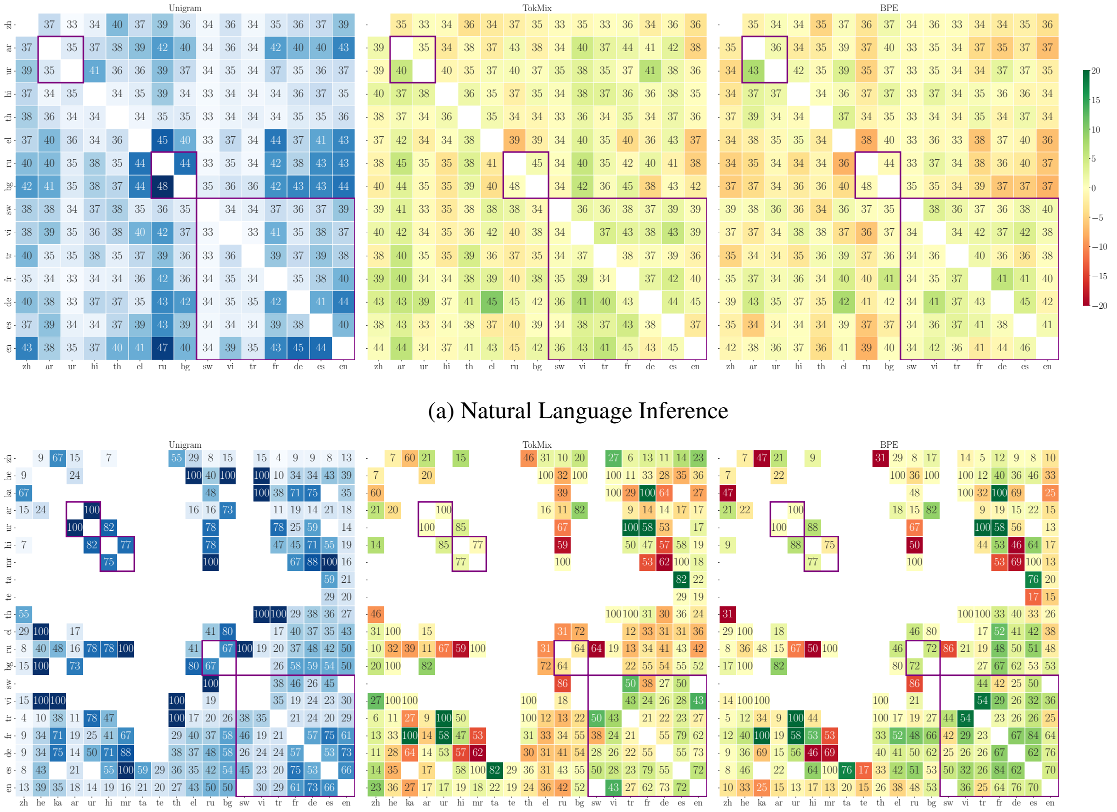


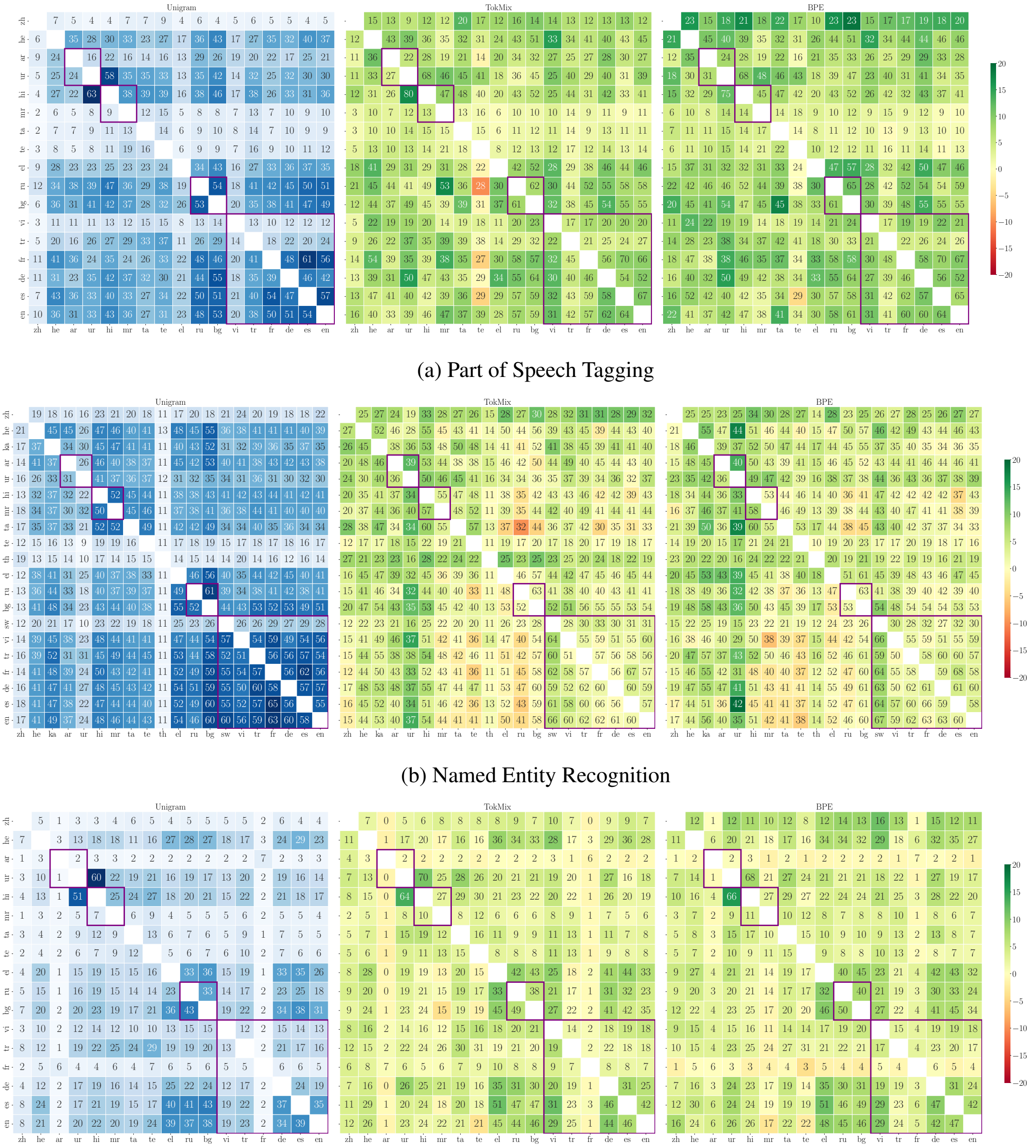
Figure 8: Jensen-Shanon divergence for three tokenization methods, computed on 20 languages.



(a) Natural Language Inference

(b) Sentence Retrieval

Figure 9: Cross-lingual transfer for the sentence-level tasks for 20 languages. The absolute values are presented for the Unigram tokenizer. For other tokenization methods, we show the difference from the unigram algorithm.



(a) Part of Speech Tagging

(b) Named Entity Recognition

(c) Dependency labeling

Figure 10: Cross-lingual transfer for the token-level tasks on 20 languages. The absolute values are presented for the Unigram tokenizer. For other tokenization methods, we show the difference from the unigram algorithm.

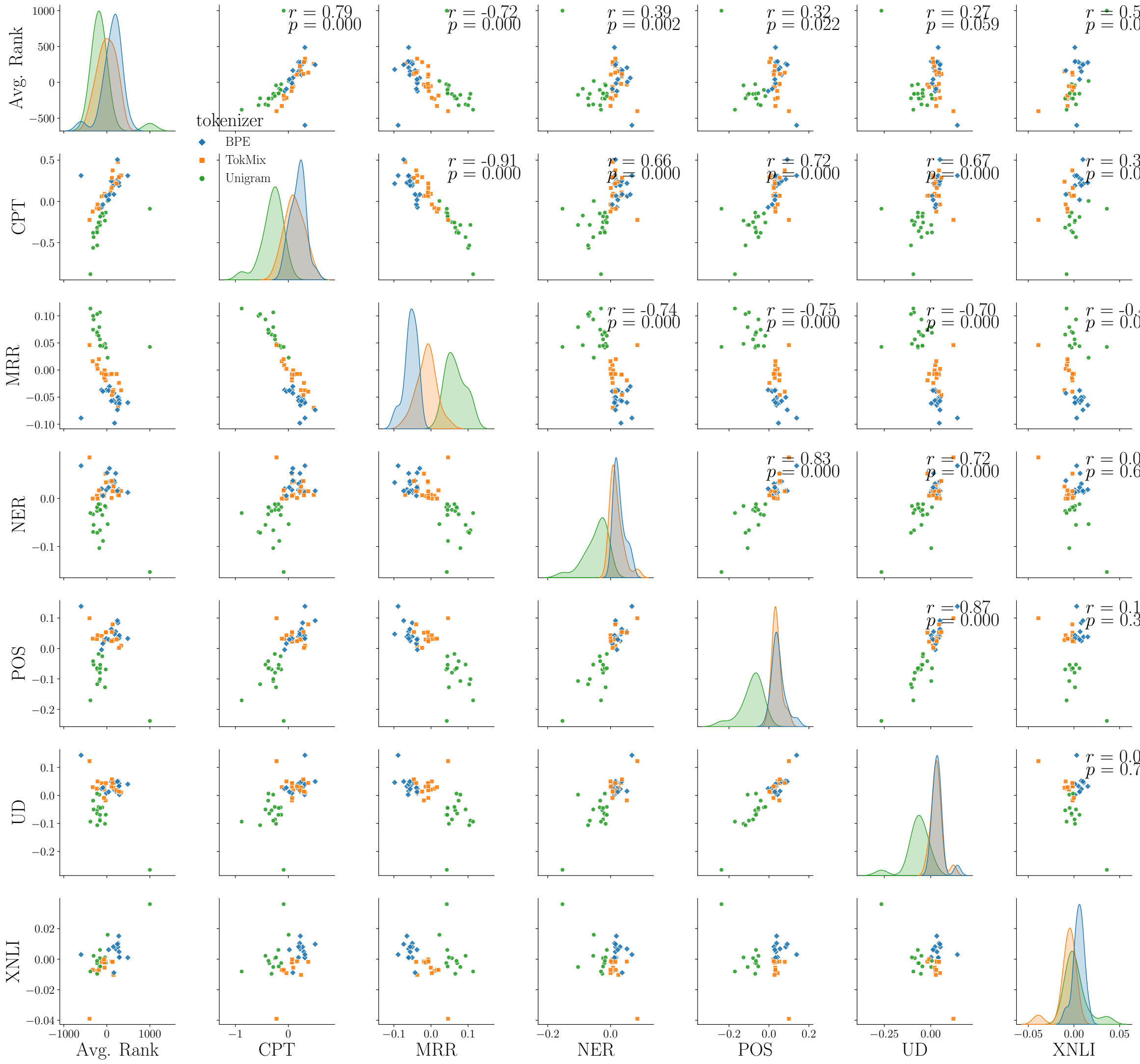


Figure 11: Correlation analysis for pairs of factors: *vocabulary overlap* metrics, language modeling performance (MRR), and downstream tasks. The diagonal of the figure presents the density of distribution of each feature. The results are grouped by the type of tokenizer applied. Analysis was done in 20 language setting. In the top right corner of each sub-plot, we show Spearman correlation coefficient and associated p-value.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Yes, section Limitations.

☑ A2. Did you discuss any potential risks of your work?
Yes. we cannot think of many risks. One of possible risks might lie in under-representing the low-resource languages but actually we propose possible improvements on that.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Yes. Section 1

☑ A4. Have you used AI writing assistants when working on this paper?
For reformulation of our English text (this does not need to be disclosed) we also report the use of coding assistants in the README we never use long coding assistant suggestions in verbatim and check the outputs closely.

B ☑ Did you use or create scientific artifacts?

> Yes, we use SentencePiece, Huggingface - cited in section 2 and appendix. Datasets used are cited in section 4.2. We do not publish the models we trained but we publish the code to reproduce the results along with a metric-computation utility package

☑ B1. Did you cite the creators of artifacts you used?
Yes, in the same sections as above

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
We do not discuss the license terms in the paper. The libraries we use are licensed under the Apache 2.0 license which allows the use of the tools for research. The datasets are released for public use.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No, We don't discuss the use of the existing artifacts, nevertheless our use is consistent with their intended use. We will specify the license under which we release our code.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. We use existing, publicly released datasets. We therefore assume that these steps were already taken.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Yes, Section 4.2.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Yes, Section 4.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

Yes, Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Yes, Appendix A.1

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Yes, Appendix A.1

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Yes, Section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Yes, Appendix A.1

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.