An Empirical Study of Multilingual Vocabulary for Neural Machine Translation Models

Kenji Imamura and Masao Utiyama
National Institute of Information and Communications Technology,
Seika-cho, Kyoto, 619-0289, Japan
{kenji.imamura, mutiyama}@nict.go.jp

Abstract

In this paper, we discuss multilingual vocabulary for neural machine translation models. Multilingual vocabularies should generate highly accurate machine translations regardless of the languages, and have preferences so that tokenized strings contain rare out-of-vocabulary (OOV) tokens and token sequences are short. In this paper, we discuss the characteristics of various multilingual vocabularies via tokenization and translation experiments. We also present our recommended vocabulary and tokenizer.

1 Introduction

In recent tasks that use neural models, including neural machine translation, we usually fine-tune pretrained models (e.g., Devlin et al. (2019); Liu et al. (2020)). When a pretrained model is fine-tuned, the training corpora are different from those used for pretraining, in which the vocabulary must be different. However, pretrained models determine their vocabulary in advance, and it is difficult to change the vocabulary during fine-tuning. Therefore, it is important to discuss the first vocabulary.[1]

On the other hand, it becomes common to process multiple languages in machine translation and large language models (LLMs) because neural models can be packed multiple languages into a model (e.g., Johnson et al. (2017)). In this paper, we discuss vocabularies appropriate for multilingual neural models. The target task is machine translation that uses encoder-decoder models. Our aim is to decide the vocabulary that is suitable for our multilingual translation models.

Figure 1 illustrates the typical structure of an encoder-decoder model (Vaswani et al., 2017). In this structure, there are five modules related to vocabulary: 1) source tokenizer, 2) target tokenizer,

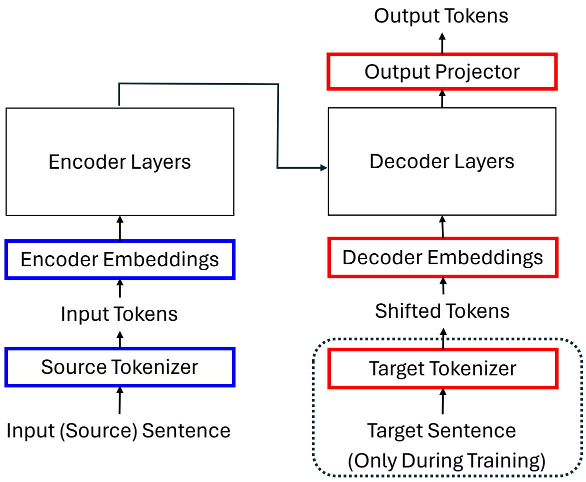


Figure 1: Vocabulary-related modules in an encoder-decoder model.

3) encoder embeddings, 4) decoder embeddings, and 5) output projector. The tokenizers tokenize a string into tokens, which consist of (sub-)words in the vocabulary of each tokenizer, except for out-of-vocabulary (OOV) strings. Neural models convert them into dense representations by looking up the tokens in the word embedding tables. Thus, the vocabularies in the tokenizers and neural model (the embedding tables and output projector) are essentially identical. It is possible to use different vocabularies between the encoder and decoder. However, shared vocabulary is generally used in multilingual models because both input and output strings are multilingual (e.g., Liu et al. (2020); Fan et al. (2020)). In this paper, we assume that the vocabularies of the above five modules are identical, unless otherwise specified.

We suppose that the preferences or requirements of the multilingual vocabulary for neural models are as follows.

1. High accuracy is preferred in target tasks. Because we use the machine translation task in this paper, high translation quality is pre-

[1] It is possible to only add words in the vocabulary (Tang et al., 2020; Imamura and Sumita, 2022).

Proceedings of the Eleventh Workshop on Asian Translation (WAT 2024), pages 22–35
November 16, 2024

ferred.

2. Token sequences, into which arbitrary strings are tokenized using the vocabulary, do not contain OOV tokens. This is a high preference because the OOV tokens certainly reduce the accuracy of tasks (Sennrich et al., 2016).

3. Token sequences are short (i.e., the numbers of tokens are small) because, generally, the shorter the input, the better the output (Arivazhagan et al., 2019).

4. Small models (i.e., the number of model parameters is small) are better for computation during training and inference. The number of parameters in the word embedding tables increases in proportion to the vocabulary size and accounts for a large portion in neural models. Therefore, a small vocabulary size is better from the viewpoint of the number of model parameters. However, it results in longer token sequences, and a tradeoff emerges between it and a preference for No. 3. We determine the balance of the two preferences using translation quality.

5. Regardless of the languages, strings with the same meaning are tokenized into similar numbers of tokens. We presume that this preference reduces complexity during translation.

6. The token sequences can be read by humans. Although this preference does not affect translation quality, high readability is better for debugging by humans.

In this paper, we discuss the vocabularies that satisfy the above preferences for multilingual models, which manage a mixture of various script types. Note that we consider No. 1 to be the most important preference, the second preference is No. 2, and the remaining preferences are optional.

The remainder of this paper is organized as follows: In Section 2, we explain related work, which includes studies of multilingual models. Next, we discuss preferred vocabulary via tokenization and translation experiments in Sections 3 and 4, respectively. In Section 5, we compare our experimental results with findings of conventional vocabulary studies, and we conclude the paper in Section 6.

2 Related Work

2.1 Multilingual Models

Table 1 shows the list of major multilingual (partially monolingual) models and their vocabularies/tokenizers.

Multilingual BERT (mBERT) (Devlin et al., 2019) and XLM-RoBERTa (XLM-R) (Conneau et al., 2020) are categorized as multilingual encoder models. These encoder models are applied to various natural language understanding tasks.

For encoder-decoder models, which are used for machine translation, multilingual BART (mBART) (Liu et al., 2020; Tang et al., 2020), M2M-100 (Fan et al., 2020), NLLB-200 (NLLB Team et al., 2022), and mT5 (Xue et al., 2021) are categorized as the multilingual models. Note that mBART and XLM-R use the same tokenization model.

Recent LLMs are resultantly multilingual, even though they learn using English Web text, because they contain other languages. Their vocabulary sizes are rather small: the size of GPT2 (Radford et al., 2019) is 50K and that of LlaMa2 (Touvron et al., 2023) is 32K.

Many multilingual models use SentencePiece (Kudo and Richardson, 2018) as their tokenizers. In this paper, we use SentencePiece for our experiments. Note that byte pair encoding (BPE) (Sennrich et al., 2016) and unigram models (Kudo, 2018) are known as major subword encoding methods. We use the unigram models in this paper.

2.2 Byte-level BPE / Byte Fallback

If an input string contains OOV characters, there are two behaviors of tokenizers (Table 2).

1) The tokenizer decomposes the OOV parts into characters. In this case, the word embeddings become unknown (indicated by `<UNK>`).

2) The tokenizer decomposes the OOV parts into byte sequences (Radford et al., 2019). This method is called byte-level BPE in the byte-pair encoding and byte fallback in SentencePiece. They assume that input strings are encoded in UTF-8. If the vocabulary of the neural models includes all bytes (256 bytes), no OOV tokens occur. However, readability decreases because humans cannot understand the string. Additionally, the decoder may generate invalid byte sequences that are

Type	Model	Tokenizer	#Langs.	Vocab. size	Byte fallback
Encoder only	mBERT	WordPiece (Schuster and Nakajima, 2012)	104	120K	
	XLM-R†	SentencePiece/Unigram (Kudo and Richardson, 2018)	100	250K	
Encoder-decoder	mBART†	SentencePiece/Unigram	100	250K	
	M2M-100	SentencePiece/BPE	100	128K	
	NLLB-200	SentencePiece/BPE	200	256K	
	mT5	SentencePiece/Unigram	101	250K	✓
Decoder only	GPT2	Byte-level BPE (Radford et al., 2019)	1	50K	✓
	LlaMa2	SentencePiece/BPE	1+‡	32K	✓

Table 1: Tokenizer and vocabulary of major multilingual models. †XLM-R and mBART use the same tokenizer with the same vocabulary. ‡ 90% of the training corpus of LlaMa2 is in English, and the rest is multilingual.

Method	Example
Source	群衆が集結しました。
1) Character	群␣<UNK>␣が␣集␣結␣しました␣。
2) Byte fallback	群␣<0xE8>␣<0xA1>␣<0x86>␣が␣集␣結␣しました␣。

Table 2: Example of byte fallback. Japanese character '衆' is fallbacked if it is not contained in the vocabulary.

not decoded into UTF-8 if byte fallback is applied to the decoder. The detokenizer must address this problem.

We also confirm the effects of byte fallback.

2.3 Flores+ Dataset

The Flores+ dataset (NLLB Team et al., 2022; Goyal et al., 2021)[2] is an evaluation dataset that covers 200 languages. It was created by translating sentences that were sampled from articles in English Wikinews, Wikijounior, and Wikivoyage into other languages. Therefore, the sentences are parallel among languages other than English. A total of 997 and 1,012 sentences are published as the development (dev) and development-test (devtest) sets, respectively.[3]

The dataset contains the language and its script type in the filenames. We use the categories (language names and script types) of Flores+ in this paper.

3 Tokenization Experiments

In this section, we evaluate tokenization using various vocabularies/tokenizers. We evaluate translation in Section 4.

3.1 Experimental Settings

Target Languages We selected 98 languages (26 script types) from the Flores+ dataset for which there were more than 100K lines in the CC-100 corpus (a set of monolingual corpora) (Conneau et al., 2020; Wenzek et al., 2020).

Considering the script types of Flores+, 55 out of 98 languages use a Latin script, such as English, and 20 languages use scripts unique to each language, such as Greek, (simplified and traditional) Chinese, Japanese, and Thai. The list of languages and script types is shown in Table 6 in Appendix A.

Tokenizer/vocabulary We evaluated M2M-100, XLM-R/mBART, NLLB-200, mT5, and LlaMa2 for existing models. For our original models, we evaluated unigram models of SentencePiece learned under various conditions.

Training Corpus for SentencePiece We randomly selected the training sets for each language from the CC-100 corpus.[4] We selected 20 million lines in total. The mean number of lines was approximately 200 thousand per language, but we controlled the sampling size using a temperature coefficient, as we describe later.

Other Settings for SentencePiece We used 0.9995 for character coverage, and the number of seed pieces was 100 times the vocabulary size.

Evaluation We evaluated the tokenization results of the 98 devtest sets in Flores+ using the following metrics.

[2] https://github.com/openlanguagedata/flores

[3] The test set is not published.

[4] The largest set in CC-100 is 1.8 billion lines of English and the smallest set is 120 thousand lines of Lingala.

- average number of tokens and variance (standard deviation) for all languages.
- total number of OOV tokens.
- number of fallbacked bytes when we applied byte fallback.

We preferred a small number of tokens (i.e., short token sequences) and a small number of OOV tokens. The low variance of the number of tokens indicated that sentences with the same meaning were tokenized in close number of tokens, regardless of the languages.

Comparison Methods We compared tokenizers/vocabularies under various conditions as follows.

- **Vocabulary Size:**
  We compared the vocabulary sizes 250K and 64K (or 100K). The vocabulary size affects the length of token sequences and neural model size.

- **Byte Fallback:**
  We compared cases with and without byte fallback. This condition influences the number of OOV tokens.

- **Additional Characters:**
  We added approximately 52K characters, which are U+0000 to U+D7FF in the basic multilingual plane of Unicode and have character names in the Python unicodedata module. Adding characters to the vocabulary enables us to control OOV tokens using an alternative to byte fallback.

  Note that we can also control OOV tokens by changing the character coverage setting during SentencePiece training. In this study, we used the additional character method to control OOV tokens.

- **Language Balance:**
  How to determine the sampling size of the training corpus for each language. We evaluated the following two methods, one is based on language distribution in the corpus, and another is based on the script types. The methods changed the importance of low-resource languages and languages that use the unique scripts. Both methods control the corpus size using the inverse temperature coefficient $1/\tau$ (temperature sampling) (Lample and Conneau, 2019; Arivazhagan et al., 2019).

a) This case follows the distribution of the CC-100 corpus (hereafter, 'Corpus'). This means that the size of high-resource languages becomes large. The training corpus size s_l of language l is determined by the following equation.

$$s_l \propto \left(\frac{c_l}{\sum_i^L c_i} \right)^{1/\tau}, \quad (1)$$

where c_l denotes the number of CC-100 lines of language l, and L denotes the number of languages ($= 98$).

b) This case uses the script types (hereafter, 'Script'). The training size of each language is uniform for a script type. Smoothing is based on the number of languages in a script type. The size of the languages that use unique scripts becomes large and that of the languages using the Latin script becomes small even though we apply temperature sampling.

$$s_l \propto \frac{(1/ns_l)^{1/\tau}}{\sum_i^L (1/ns_i)^{1/\tau}}, \quad (2)$$

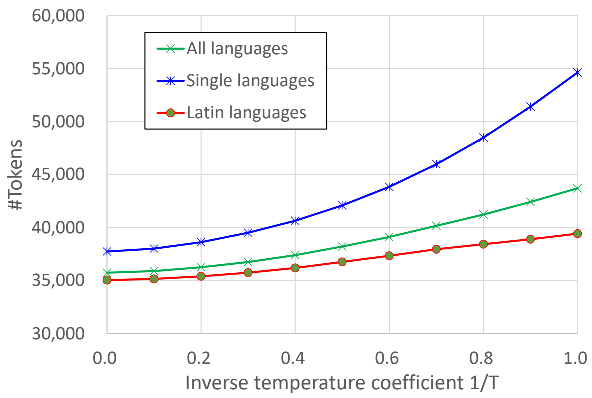
where ns_l denotes the number of languages in the script type to which language l belongs (e.g., 55 languages belong to the Latin script type, and one language belongs to the Japanese script type).

3.2 Result 1: Language Balance

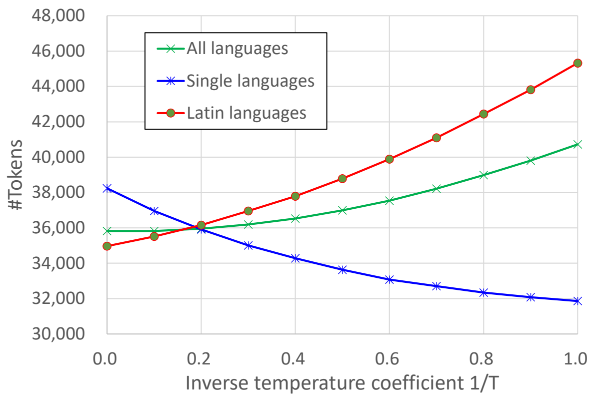
Before the comparison experiments, we determined the optimal inverse temperature coefficient $1/\tau$ by changing the training corpus size for SentencePiece. We evaluated the 250K vocabulary with byte fallback without additional characters.

Figure 2 shows the change of the number of tokens in the Flores+ devtest set when we changed the inverse temperature coefficient from 0.0 to 1.0. It includes the average of all languages, the average of the languages that use the unique script (20 languages; represented as 'Single' languages), and the average of the languages of the Latin script (55 languages; 'Latin' languages).

a) When we used the Corpus method, the number of tokens and the difference between the Single and Latin languages became the smallest when $1/\tau = 0.0$.



a) Corpus: Using the distribution of CC-100.



b) Script: Using the script types.

Figure 2: Number of tokens of Flores+ according to the inverse temperature coefficient $1/\tau$.

b) When we used the Script method, the number of tokens in the Latin languages increased as $1/\tau$ increased. Conversely, that of the Single languages decreased as $1/\tau$ increased and they were balanced when $1/\tau = 0.2$.

These results show that it was effective to balance languages by changing the training corpus size of each language using the inverse temperature coefficient. In subsequent experiments, we used the optimal inverse temperature coefficient that balanced all languages, that is, the standard deviation of the number of tokens became the smallest.

3.3 Result 2: Tokenization

Table 3 shows the tokenization results for the Flores+ devtest set using various tokenizers and vocabularies. 'Avg. #tokens' is the average number of tokens in all languages, and its standard deviation indicates the variance among languages. If the standard deviation is small, differences among languages must also be small. '#OOV' indicates the total number of OOV tokens, and '#Fallbacked bytes' is the total number of fallbacked bytes in all languages.

First, we confirmed the tokenization results of the baselines. The mBART/XLM-R and NLLB-200 tokenizers generated the least number of tokens, and mBART/XLM-R generated the least OOV tokens of the two tokenizers, even though it does not use byte fallback. From the viewpoint of OOV tokens, mT5, which uses byte fallback, was the best; however, the number of tokens was more than that of mBART/XLM-R. We consider that mBART/XLM-R was the most suitable tokenizer/vocabulary for our preferences (c.f., Section 1).

Next, we compared our SentencePiece unigram models, referring to the preferences. We confirm the translation quality in the next section.

First, the number of OOV tokens became zero using byte fallback.

The average number of tokens was most affected by the vocabulary size. The tokenizers with the 250K vocabulary became a similar number of tokens regardless of the other conditions. Although not shown in the table, the vocabulary size also affected the number of model parameters. When we used a Transformer big model (Vaswani et al., 2017), the number of model parameters was approximately 430 million for the 250K vocabulary and 240 million for the 64K vocabulary. The vocabulary size is a trade-off between the number of tokens and the number of parameters, and we determined the optimal size using translation quality.

The standard deviation of the number of tokens indicates the variance of languages. However, it was less affected by the language balance and byte fallback because all deviations of the 250K tokenizers were less than 3,700. It was most influenced by the size of the training corpus, as shown in Section 3.2.

Finally, focusing on the number of fallbacked bytes, the number decreased when there were additional characters. For example, 5,848 bytes in `250K_S+B` decreased to 48 bytes in `250K_S+B+C52K`. Adding characters is a solution to improve readability if translation quality is the same.

Tokenization examples of several languages are shown in Tables 8 to 10 in Appendix C.

Tokenizer/ vocabulary	Vocab. size	Byte fallback	Additional characters	Lang. balance	Avg. #tokens (std. dev.)	#OOV	# Fallbacked bytes
Baselines							
M2M-100	128K			Corpus	42,196 (8,542)	38,942	N/A
mBART/XLM-R	250K			Corpus	37,632 (6,246)	30	N/A
NLLB-200	256K			Corpus	37,579 (4,900)	16,739	N/A
mT5	250K	√		Corpus	45,365 (9,979)	0	81
LlaMa2	32K	√		–[5]	96,836 (75,630)	0	2,989,581
SentencePiece/Unigram							
250K_C+B	250K	√	0	Corpus	35,562 (3,510)	0	11,601
250K_S	250K		0	Script	35,900 (3,422)	1,873	N/A
250K_S+B	250K	√	0	Script	35,948 (3,367)	0	5,848
250K_S+B+C52K	250K	√	52K	Script	37,095 (3,602)	0	48
64K_S+B	64K	√	0	Script	45,504 (4,294)	0	4,745
100K_S+B+C52K	100K	√	52K	Script	47,410 (4,676)	0	48

Table 3: Tokenization results. The tokenizer/vocabulary names of SentencePiece are combinations of the vocabulary size, language balance ('C' and 'S' represent 'Corpus' and 'Script,' respectively), byte fallback ('B'), and additional characters (C52K).

4 Translation Experiments

4.1 Experimental Settings

We evaluated the translation quality as follows:

Tokenizer/vocabulary From the tokenizers/vocabularies used in Section 3, we selected all SentencePiece vocabularies and mBART/XLM-R and mT5 as the baselines.

Translation Languages We selected the following eight out of 98 languages and trained a multilingual translation model in all directions ($8 \times 7 = 56$ directions) for each vocabulary:

- **Latin Languages:**
  English, Spanish, and Vietnamese: We selected one European language and one Asian language other than English.
- **Single Languages:**
  Japanese and Mandarin Chinese (Standard Beijing): Although their characters have the same origin, they use different glyphs (i.e., different character codes), in most cases.
- **Other Languages:**
  Modern Standard Arabic, Hindi, and Russian: These are the other script types of the above languages.

Parallel Corpus We sampled 1 million sentences for each language pair from the NLLB-200 corpus as the parallel corpus to train the translation models. We sampled sentences independently for each language pair. Therefore, the importances of the languages are the same in this experiment.

Translation Models We used the Transformer big models (Vaswani et al., 2017) (1,024 embedding and 4,096 FFN dimensions, six layers for the encoder and decoder) implemented by FairSeq (Ott et al., 2019), and learned multilingual models in 56 (8×7) directions. Like the M2M-100 model (Fan et al., 2020), the multilingual models were trained while we supplied language tags (e.g., '`__en__`' for English) at the head of the source and target sentences.

Hyperparameters The details of the hyperparameters are shown in Appendix B.

Evaluation We evaluated the translation quality using the average scores of 56 directions of ChrF++ (Popović, 2017) and COMET (Rei et al., 2022) (using the `wmt22-comet-da` model) implemented in SacreBLEU (Post, 2018). For the statistical test, we used binomial testing with 56 trials, in which a trial indicated a direction ($p < 0.05$).

4.2 Results

Table 4 shows the translation quality for each vocabulary. Among all vocabularies, mBART/XLM-R achieved the highest scores. This is because it contained (not zero, but) very few OOV tokens and the number of tokens was low.

Next, we focused on the results of our SentencePiece unigram models. Regarding the vocabulary size, the translation qualities of the 250K

[5]This vocabulary does not balance languages because the model is not precisely multilingual.

Tokenizer/vocabulary	Vocab. size	Byte fallback	Additional characters	Lang. balance	Avg. score ChrF++	COMET
Baselines						
XLM-R/mBART	250K			Corpus	**41.13**	**.8237**
mT5	250K	√		Corpus	40.44	.8176
SentencePiece/Unigram						
250K_C+B	250K	√	0	Corpus	<u>40.93</u>	.8211
250K_S	250K		0	Script	40.72	.8167
250K_S+B	250K	√	0	Script	<u>40.93</u>	<u>.8212</u>
250K_S+B+C52K	250K	√	52K	Script	40.89	.8208
64K_S+B	64K	√	0	Script	40.21	.8139
100K_S+B+C52K	100K	√	52K	Script	40.09	.8127

Table 4: Translation quality for each tokenizer/vocabulary. The tokenizer/vocabulary names of SentencePiece consisted of the vocabulary size, language balance ('C' is Corpus, and 'S' is Script Type), byte fallback (B), and additional characters (C52K). The bold scores indicate the highest score, and the underlined scores indicate the second-best scores.

vocabularies were better than those of the 64k (or 100K) sizes. For example, the ChrF++ and COMET scores of `250K_S+B` were higher than those of `64K_S+B` ($p = 1.6 \times 10^{-15}$), and the scores of `250K_S+B+C52K` were higher than those of `100K_S+B+C52K` ($p = 5.6 \times 10^{-17}$).

The translation quality with byte fallback was significantly higher than that without byte fallback when comparing `250K_S` and `250K_S+B` ($p = 2.5 \times 10^{-5}$), even though the difference was small.

Regarding additional characters, although we could not find a significant difference between `250K_S+B` and `250K_S+B+C52K`, the scores of `64K_S+B` were significantly higher than those of `100K_S+B+C52K` ($p = 6.1 \times 10^{-4}$). Additional characters were not effective. We suppose that this was because multi-character subwords reduced in the vocabulary or characters that were not learned remained when we added 52K characters.

4.3 When Different Vocabulary Sizes are Used between the Encoder and Decoder

In the preceding discussion, we assumed that a shared vocabulary was used in the encoder and decoder. However, the optimal vocabularies of the encoder and decoder may not be the same because the encoder is responsible for natural language understanding, and the decoder is responsible for generation. Therefore, in this subsection, we confirm the translation quality if we change the vocabulary between the encoder and decoder.

Specifically, we performed a translation experiment by changing the vocabulary sizes between the encoder and decoder. We used `250K_S+B` and `64K_S+B` (i.e., the language balance was the script type, with byte fallback, and the vocabulary sizes were 250K and 64K).

Vocab. size Encoder	Decoder	ChrF++	COMET
250K		**40.93**	**.8212**
250K	64K	40.73	.8192
64K	250K	40.82	.8203
64K		40.21	.8139

Table 5: Translation quality (average scores in the 56 directions) when changing the vocabulary sizes of the encoder and decoder.

Table 5 shows the result. Regardless of whether we changed the vocabulary size of the encoder or decoder, the scores were intermediate between those of 250K and 64K. The shared vocabulary of the encoder and decoder was suitable to achieve high translation quality.

5 Comparison with Conventional Vocabulary Studies

There have been various vocabulary studies using multilingual neural models. The findings of these studies, in comparison with the results of our study, can be summarized as follows:

Arivazhagan et al. (2019) built multilingual models covering 103 languages using various conditions. They also investigated vocabularies for the models and reported the following findings.

1. Translation quality is better when a large vocabulary is used.
2. Changes to the language balance of the vocabulary using temperature sampling do not

significantly affect translation quality.

In our experiments, a large vocabulary resulted in better translation quality. In addition, the language balance was not observed to have a significant effect on quality.

Gowda and May (2020) investigated the optimal vocabulary size for multiple languages (using only single-directional translation models). They reported that the optimal vocabulary size depends on the training corpus size for the translation models. Namely, a large vocabulary is better in high-resource languages and a small vocabulary is preferable low-resource languages. As described in Section 4, our experiments indicate that a large vocabulary is better because we use 1,000,000 parallel sentences for each direction, which is regarded as a high-resource condition.

Zhang et al. (2022) constructed multilingual vocabularies for eight languages with different English ratios in the training corpora, and investigated the impact on the translation quality. In addition to the findings of conventional studies, they investigated the effects of byte fallback and showed that this feature does not significantly affect the translation quality. In our experiments, in addition to eliminating OOV tokens, byte fallback was found to enhance the translation quality. Therefore, we consider it preferable to use byte fallback.

6 Conclusions

In this paper, we discussed multilingual vocabulary for neural machine translation models. Our findings are summarized as follows:

1. Among all vocabularies, mBART/XLM-R was the best in the machine translation task. Although the tokenizer of mBART/XLM-R did not use byte fallback, the number of OOV tokens was small and, consequently, the translation quality became high.

   Among the vocabularies of our SentencePiece models, the vocabularies of 250K with byte fallback achieved high quality.

2. Byte fallback was effective for eliminating OOV tokens, and the translation quality was better than that without byte fallback.

3. The vocabularies of the 250K size generated the smallest number of tokens (the shortest length of token sequences). These vocabularies had the disadvantage that the number of model parameters increased. However, translation quality was better than that for the 64K vocabulary.

4. To tokenize multilingual sentences into a similar (close) number of tokens, it was effective to control the training data size of each language. It could be controlled using a temperature coefficient.

5. Readability increased when the number of fallbacked bytes was low. However, translation quality decreased when we increased character coverage by adding characters into the vocabulary.

Recommended Vocabulary/Tokenizer Based on the vocabulary of mBART/XLM-R, we recommend using a tokenizer with byte fallback. In future work, we will build multilingual translation models using the multilingual vocabulary discussed in this paper.

Limitations

The results in this paper were a case study because our experiments were not comprehensive.

Ethics Statement

Our vocabularies were created automatically from corpora, and we did not check the contents. Therefore, they may contain inappropriate words.

Acknowledgments

Part of this work was conducted under the commissioned research program 'Research and Development of Advanced Multilingual Translation Technology' in the 'R&D Project for Information and Communications Technology (JPMI00316)' of the Ministry of Internal Affairs and Communications (MIC), Japan.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *Preprint*, arXiv:1907.05019.

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

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.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *arXiv e-print*, 2010.11125. *arXiv preprint*.

Thamme Gowda and Jonathan May. 2020. Finding the optimal vocabulary size for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3955–3964, Online. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation.

Kenji Imamura and Eiichiro Sumita. 2022. Extending the subwording model of multilingual pretrained models for new languages. *Preprint*, arXiv:2211.15965.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

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 (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

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

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NIPS-2019)*, pages 7059–7069.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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. *arXiv e-print*, 2207.04672. *Preprint*, arXiv:2207.04672.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. 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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André

F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.

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.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv e-print*, 2008.00401. *Preprint*, arXiv:2008.00401.

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. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

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

Shiyue Zhang, Vishrav Chaudhary, Naman Goyal, James Cross, Guillaume Wenzek, Mohit Bansal, and Francisco Guzman. 2022. How robust is neural machine translation to language imbalance in multilingual tokenizer training? In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 97–116, Orlando, USA. Association for Machine Translation in the Americas.

A Language List in this Paper

Table 6 shows the list of 98 languages used in this paper, which is organized by script type.

B Hyperparameters for Translation Experiments

Table 7 shows the list of hyperparameters used in the experiments in Section 4.

C Tokenization Examples

Tables 8 to 10 show tokenization examples, in which the same sentences (or translations) obtained from the Flores+ dev set were tokenized by each tokenizer. Depending on the tokenizers, the number of tokens vary significantly in a language, and each tokenizer has strong and weak languages. Among the tokenizers, `mBART/XLM-R` and `250K_S+B` tokenized the sentences into fewer tokens on average.

Script type	#Langs.	Languages
Arabic	6	**Modern Standard Arabic**, Southern Pashto, Western Persian, Sindhi, Urdu, Uyghur
Armenian	1	Armenian
Bengali	2	Assamese, Bengali
Cyrillic	9	Belarusian, Bulgarian, Kazakh, Kyrgyz, Macedonian, Halh Mongolian, **Russian**, Serbian, Ukrainian
Devanagari	4	**Hindi**, Marathi, Nepali, Sanskrit
Ge'ez	1	Amharic
Georgian	1	Georgian
Greek	1	Greek
Gujarati	1	Gujarati
Gurmukhi	1	Eastern Panjabi
Hebrew	2	Hebrew, Eastern Yiddish
Hungul	1	Korean
Japanese	1	**Japanese**
Kannada	1	Kannada
Khmer	1	Khmer
Lao	1	Lao
Latin	55	Afrikaans, Tosk Albanian, North Azerbaijani, Basque, Norwegian Bokmål, Bosnian, Catalan, Haitian Creole, Croatian, Czech, Danish, Dutch, **English**, Esperanto, Estonian, Finnish, French, Scottish Gaelic, Galician, Ganda, German, Hausa, Hungarian, Icelandic, Igbo, Indonesian, Irish, Italian, Javanese, Northern Kurdish, Standard Latvian, Lingala, Lithuanian, Plateau Malagasy, Standard Malay, West Central Oromo, Polish, Portuguese, Romanian, Slovak, Slovenian, Somali, **Spanish**, Sundanese, Swahili, Swedish, Tagalog / Filipino, Tswana, Turkish, Northern Uzbek, **Vietnamese**, Welsh, Wolof, Xhosa, Zulu
Malayalam	1	Malayalam
Myanmar	1	Burmese
Odia	1	Odia
Simplified Chinese	1	**Mandarin Chinese (Standard Beijing)**
Sinhala	1	Sinhala
Tamil	1	Tamil
Telugu	1	Telugu
Thai	1	Thai
Traditional Chinese	1	Mandarin Chinese (Taiwanese)
Total	98	

Table 6: Script types and languages. #Langs. indicates the number of languages. The languages in bold were used in the translation experiments.

Type	Name	Setting
Model	Architecture	Transformer big
	Embedding dimension	1,024
	FFN inner dimension	4,096
Training	Dropout	0.3
	Loss function	Label smoothed cross-entropy
	Label smoothing	$\epsilon = 0.1$
	Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98$)
	Learning rate	5e-4
	LR scheduler	Inverse square root
	Warm-up steps	4,000
	Global batch size	Roughly 128,000 tokens
	Early Stopping	No-update 9 epochs
Test	Beam width	10

Table 7: Hyperparameters for the translation experiments.

Tokenizer/	English		Spanish		Vietnamese	
vocabulary	#Tokens	Sample	#Tokens	Sample	#Tokens	Sample
M2M-100	15	▁Local ␣ ▁media ␣ ▁reports ␣ ▁an ␣ ▁air ␣ port ␣ ▁fire ␣ ▁vehicle ␣ ▁roll ␣ ed ␣ ▁over ␣ ▁while ␣ ▁ respond ␣ ing ␣ .	23	▁La ␣ ▁prensa ␣ ▁local ␣ ▁inform ␣ ó ␣ ▁que ␣ ▁una ␣ ▁patr ␣ ulla ␣ ▁de ␣ ▁bom ␣ ber ␣ os ␣ ▁del ␣ ▁aerop ␣ uerto ␣ ▁vol ␣ có ␣ ▁ mientras ␣ ▁presta ␣ ba ␣ ▁ servicio ␣ .	19	▁Truyền ␣ ▁thông ␣ ▁địa ␣ ▁ phương ␣ ▁đưa ␣ ▁tin ␣ ▁một ␣ ▁ phương ␣ ▁tiện ␣ ▁chữa ␣ ▁cháy ␣ ▁sân ␣ ▁bay ␣ ▁đã ␣ ▁tới ␣ ▁ khi ␣ ▁trả ␣ ▁lời ␣ .
mBART/XLM-R	14	▁Local ␣ ▁media ␣ ▁reports ␣ ▁an ␣ ▁airport ␣ ▁fire ␣ ▁vehicle ␣ ▁ rolle ␣ d ␣ ▁over ␣ ▁while ␣ ▁ respond ␣ ing ␣ .	20	▁La ␣ ▁prensa ␣ ▁local ␣ ▁inform ó ␣ ▁que ␣ ▁una ␣ ▁patru ␣ lla ␣ ▁de ␣ ▁bombe ␣ ros ␣ ▁del ␣ ▁ aeropuerto ␣ ▁vol ␣ có ␣ ▁ mientras ␣ ▁presta ␣ ba ␣ ▁ servicio ␣ .	19	▁Truyền ␣ ▁thông ␣ ▁địa ␣ ▁ phương ␣ ▁đưa ␣ ▁tin ␣ ▁một ␣ ▁ phương ␣ ▁tiện ␣ ▁chữa ␣ ▁cháy ␣ ▁sân ␣ ▁bay ␣ ▁đã ␣ ▁tới ␣ ▁ khi ␣ ▁trả ␣ ▁lời ␣ .
NLLB-200	14	▁Local ␣ ▁media ␣ ▁reports ␣ ▁an ␣ ▁airport ␣ ▁fire ␣ ▁vehicle ␣ ▁ rol ␣ led ␣ ▁over ␣ ▁while ␣ ▁ respond ␣ ing ␣ .	22	▁La ␣ ▁prensa ␣ ▁local ␣ ▁inform ␣ ó ␣ ▁que ␣ ▁una ␣ ▁patr ␣ ulla ␣ ▁de ␣ ▁bom ␣ beros ␣ ▁del ␣ ▁ aerop ␣ uerto ␣ ▁vol ␣ có ␣ ▁ mientras ␣ ▁presta ␣ ba ␣ ▁ servicio ␣ .	20	▁Tru ␣ yền ␣ ▁thông ␣ ▁địa ␣ ▁ phương ␣ ▁đưa ␣ ▁tin ␣ ▁một ␣ ▁ phương ␣ ▁tiện ␣ ▁chữa ␣ ▁cháy ␣ ▁sân ␣ ▁bay ␣ ▁đã ␣ ▁tới ␣ ▁ khi ␣ ▁trả ␣ ▁lời ␣ .
mT5	16	▁Local ␣ ▁media ␣ ▁ ␣ reports ␣ ▁an ␣ ▁airport ␣ ▁fire ␣ ▁vehicle ␣ ▁ ␣ rolled ␣ ▁over ␣ ▁ ␣ while ␣ ▁respond ␣ ing ␣ .	25	▁La ␣ ▁ ␣ prensa ␣ ▁local ␣ ▁ inform ␣ ó ␣ ▁que ␣ ▁una ␣ ▁ patrul ␣ la ␣ ▁de ␣ ▁ ␣ bomber ␣ os ␣ ▁del ␣ ▁aero ␣ puerto ␣ ▁vol ␣ có ␣ ▁mi ␣ entras ␣ ▁presta ␣ ba ␣ ▁servicio ␣ .	38	▁Tr ␣ uyền ␣ ▁th ␣ ông ␣ ▁đ ␣ ịa ␣ ▁p ␣ hương ␣ ▁đư ␣ a ␣ ▁tin ␣ ▁m ␣ ột ␣ ▁p ␣ hương ␣ ▁ ␣ t ␣ iện ␣ ▁ch ␣ ữ ␣ a ␣ ▁chá ␣ y ␣ ▁ ␣ sân ␣ ▁bay ␣ ▁đ ␣ ã ␣ ▁ ␣ t ␣ ới ␣ ▁khi ␣ ▁tr ␣ ả ␣ ▁ ␣ l ␣ ời ␣ .
LlaMa2	14	▁Local ␣ ▁media ␣ ▁reports ␣ ▁an ␣ ▁air ␣ port ␣ ▁fire ␣ ▁vehicle ␣ ▁rolled ␣ ▁over ␣ ▁while ␣ ▁ respond ␣ ing ␣ .	27	▁La ␣ ▁pr ␣ ensa ␣ ▁local ␣ ▁ inform ␣ ó ␣ ▁que ␣ ▁una ␣ ▁patr ␣ ulla ␣ ▁de ␣ ▁bom ␣ ber ␣ os ␣ ▁del ␣ ▁aer ␣ op ␣ uerto ␣ ▁vol ␣ c ␣ ó ␣ ▁mientras ␣ ▁prest ␣ aba ␣ ▁serv ␣ icio ␣ .	53	▁Tru ␣ y ␣ ề ␣ n ␣ ▁th ␣ ô ␣ ng ␣ ▁ ␣ đ ␣ ị ␣ a ␣ ▁ph ␣ ư ␣ ơ ␣ ng ␣ ▁ ␣ đ ␣ ư ␣ a ␣ ▁tin ␣ ▁m ␣ ộ ␣ t ␣ ▁ph ␣ ư ␣ ơ ␣ ng ␣ ▁ti ␣ ệ ␣ n ␣ ▁ch ␣ ữ ␣ a ␣ ▁ch ␣ á ␣ y ␣ ▁s ␣ ân ␣ ▁bay ␣ ▁ ␣ đ ␣ ã ␣ ▁t ␣ ớ ␣ i ␣ ▁k ␣ hi ␣ ▁tr ␣ ả ␣ ▁l ␣ ờ ␣ i ␣ .
250K_S+B	16	▁Local ␣ ▁media ␣ ▁report ␣ s ␣ ▁an ␣ ▁air ␣ port ␣ ▁fire ␣ ▁ vehicle ␣ ▁rolle ␣ d ␣ ▁over ␣ ▁ while ␣ ▁respond ␣ ing ␣ .	23	▁La ␣ ▁prensa ␣ ▁local ␣ ▁inform ␣ ó ␣ ▁que ␣ ▁una ␣ ▁patru ␣ lla ␣ ▁de ␣ ▁bombe ␣ ros ␣ ▁del ␣ ▁ aero ␣ pu ␣ erto ␣ ▁vol ␣ có ␣ ▁ mientras ␣ ▁presta ␣ ba ␣ ▁ servicio ␣ .	20	▁Truyền ␣ ▁thông ␣ ▁địa ␣ ▁ phương ␣ ▁đưa ␣ ▁tin ␣ ▁một ␣ ▁ phương ␣ ▁tiện ␣ ▁chữa ␣ ▁chá ␣ y ␣ ▁sân ␣ ▁bay ␣ ▁đã ␣ ▁tới ␣ ▁ khi ␣ ▁trả ␣ ▁lời ␣ .
64K_S+B	19	▁Lo ␣ cal ␣ ▁media ␣ ▁report ␣ s ␣ ▁an ␣ ▁air ␣ port ␣ ▁fire ␣ ▁ve ␣ hi ␣ cle ␣ ▁rol ␣ led ␣ ▁over ␣ ▁while ␣ ▁respond ␣ ing ␣ .	30	▁La ␣ ▁pren ␣ sa ␣ ▁local ␣ ▁ inform ␣ ó ␣ ▁que ␣ ▁una ␣ ▁pat ␣ ru ␣ lla ␣ ▁de ␣ ▁bo ␣ mber ␣ os ␣ ▁del ␣ ▁a ␣ ero ␣ pu ␣ erto ␣ ▁vol ␣ c ␣ ó ␣ ▁ ␣ mien ␣ tras ␣ ▁ presta ␣ ba ␣ ▁servicio ␣ .	27	▁Tr ␣ uyền ␣ ▁thông ␣ ▁địa ␣ ▁ phương ␣ ▁đưa ␣ ▁tin ␣ ▁một ␣ ▁ phương ␣ ▁t ␣ iện ␣ ▁ch ␣ ữ ␣ a ␣ ▁ ␣ chá ␣ y ␣ ▁s ␣ ân ␣ ▁bay ␣ ▁đã ␣ ▁tới ␣ ▁khi ␣ ▁trả ␣ ▁l ␣ ời ␣ .

Table 8: Tokenization examples obtained from the dev set in Flores+ (1/3). The '␣' and '▁' symbols indicate the token delimiter and space character of SentencePiece, respectively.

Tokenizer/ vocabulary	Japanese #Tokens	Japanese Sample	Chinese #Tokens	Chinese Sample	Arabic #Tokens	Arabic Sample
M2M-100	27	▁⊔地⊔元⊔メ⊔ディ⊔ア⊔ の⊔報⊔道⊔によ⊔ると⊔、 ⊔空⊔港⊔の⊔消⊔防⊔車⊔ が⊔対応⊔中に⊔横⊔転⊔し た⊔ということ⊔です⊔。	21	▁当⊔地⊔媒体⊔报道⊔,⊔ 一⊔辆⊔机场⊔消⊔防⊔车 ⊔在⊔响⊔应⊔火⊔警⊔时 ⊔翻⊔了⊔车⊔。	24	▁⊔الإعلام▁⊔وسائل▁⊔أ⊔علنت▁ ⊔إ▁⊔□قلاب▁⊔عن▁⊔□مح⊔□ية الإ⊔طف⊔اء▁⊔سي⊔ارا□▁⊔حدى ⊔لإ⊔طف▁⊔توجه⊔ها▁⊔أثناء▁⊔ ⊔□ج⊔ريق▁⊔اء.
mBART/XLM-R	20	▁⊔地⊔元⊔メディア⊔の⊔ 報道⊔によると⊔、⊔空港⊔ の⊔消防⊔車⊔が⊔対応⊔中 に⊔横⊔転⊔した⊔というこ とです⊔。	17	▁⊔当地⊔媒体报道⊔,⊔一 辆⊔机场⊔消防⊔车⊔在⊔ 响应⊔火⊔警⊔时⊔翻⊔了 ⊔车⊔。	20	□محلية▁⊔الإعلام▁⊔وسائل▁⊔أعلنت▁ سيارا□▁⊔إحدى▁⊔□قلاب▁⊔عن▁⊔ ⊔توجه▁⊔أثناء▁⊔الإ⊔طف⊔اء▁⊔ ⊔□حر⊔يق▁⊔لإ⊔طف⊔اء▁⊔ها.
NLLB-200	18	▁地⊔元⊔メディア⊔の⊔報 道⊔によると⊔、⊔空港⊔ の⊔消防⊔車が⊔対応⊔中 に⊔横⊔転⊔した⊔という ことです⊔。	22	▁当⊔地⊔媒体⊔报⊔道⊔, ⊔一⊔辆⊔机⊔场⊔消防⊔ 车⊔在⊔响⊔应⊔火⊔警⊔ 时⊔翻⊔了⊔车⊔。	30	⊔الإ▁⊔وس⊔ائل▁⊔أ⊔عل⊔□ت▁ □قلاب▁⊔عن▁⊔□مح⊔□ية▁⊔ع⊔لام ⊔الإ▁⊔سي⊔ارا□▁⊔إح⊔دى▁⊔ تو⊔جهه⊔ا▁⊔أث⊔□اء▁⊔طف⊔اء ⊔□ح⊔ريق▁⊔لإ⊔طف⊔اء▁⊔.
mT5	18	▁⊔地元⊔メディア⊔の⊔報 道⊔によると⊔、⊔空港⊔の ⊔消防⊔車が⊔対応⊔中に⊔ 横⊔転⊔した⊔ということで す⊔。	18	▁⊔当地⊔媒体⊔报道⊔,⊔ 一辆⊔机场⊔消防⊔车⊔在 ⊔响应⊔火⊔警⊔时⊔翻⊔ 了⊔车⊔。	31	ال⊔إعلام▁⊔و⊔سائل▁⊔أع⊔□نت▁ ▁⊔□قلاب⊔▁⊔عن▁⊔□مح⊔□ية▁⊔ الإ⊔طف▁⊔سيارا□⊔▁⊔إح⊔دى⊔ توجه⊔ها⊔▁⊔أث⊔□اء⊔▁⊔اء⊔ ⊔□حر⊔يق▁⊔لإ⊔طف⊔اء▁⊔.
LlaMa2	48	▁⊔地⊔元⊔メ⊔デ⊔ィ⊔ア ⊔の⊔報⊔道⊔に⊔よ⊔る⊔ と⊔、⊔空⊔港⊔の⊔消⊔ <0xE9>⊔<0x98>⊔<0xB2>⊔車 ⊔が⊔<0xE5>⊔<0xAF>⊔ <0xBE>⊔<0xE5>⊔<0xBF>⊔ <0x9C>⊔中⊔に⊔<0xE6>⊔ <0xA8>⊔<0xAA>⊔<0xE8>⊔ <0xBB>⊔<0xA2>⊔し⊔た⊔ と⊔い⊔う⊔こ⊔と⊔で⊔す ⊔。	43	▁⊔当⊔地⊔<0xE5>⊔<0xAA> ⊔<0x92>⊔体⊔<0xE6>⊔ <0x8A>⊔<0xA5>⊔道⊔,⊔ 一⊔<0xE8>⊔<0xBE>⊔ <0x86>⊔机⊔场⊔消⊔<0xE9> ⊔<0x98>⊔<0xB2>⊔<0xE8>⊔ <0xBD>⊔<0xA6>⊔在⊔ <0xE5>⊔<0x93>⊔<0x8D>⊔ 应⊔火⊔<0xE8>⊔<0xAD>⊔ <0xA6>⊔时⊔<0xE7>⊔ <0xBF>⊔<0xBB>⊔了⊔ <0xE8>⊔<0xBD>⊔<0xA6>⊔ 。	78	و⊔س▁⊔أ⊔ع⊔ل⊔□⊔□⊔▁ ⊔ال⊔إ⊔ع⊔ل⊔ا▁⊔ا⊔ئ⊔ل⊔ ع▁⊔ال⊔م⊔ح⊔ل⊔ي⊔ة▁⊔م ⊔ا⊔□⊔ق⊔ل⊔ا⊔ب▁⊔□⊔ س⊔ي⊔ا▁⊔إ⊔ح⊔د⊔ى▁ ⊔ال⊔إ⊔ط⊔ف⊔ا▁⊔ر⊔ا⊔□⊔ □⊔▁⊔أ⊔ث⊔□⊔ا⊔ء▁⊔ء ل⊔إ⊔ط▁⊔و⊔ج⊔ه⊔ا⊔ ال⊔ح⊔ر⊔ي⊔ق▁⊔ف⊔ا⊔ء⊔ ⊔.
250K_S+B	20	▁⊔地⊔元⊔メディア⊔の⊔ 報道⊔によると⊔、⊔空港⊔ の⊔消防⊔車⊔が⊔対応⊔中 に⊔横⊔転⊔した⊔というこ とです⊔。	17	▁⊔当地⊔媒体报道⊔,⊔一辆 ⊔机场⊔消防⊔车⊔在⊔响应 ⊔火⊔警⊔时⊔翻⊔了⊔车⊔ 。	20	□محلية▁⊔الإعلام▁⊔وسائل▁⊔أعلنت▁ سيارا□▁⊔إحدى▁⊔□قلاب▁⊔عن▁⊔ ⊔توجه▁⊔أثناء▁⊔الإ⊔طف⊔اء▁⊔ ⊔□حر⊔يق▁⊔لإ⊔طف⊔اء▁⊔ها.
64K_S+B	28	▁⊔地⊔元⊔メ⊔ディ⊔ア⊔ の⊔報⊔道⊔による⊔と⊔、 ⊔空⊔港⊔の⊔消⊔防⊔車⊔ が⊔対応⊔中⊔に⊔横⊔転⊔ した⊔ということ⊔です⊔。	21	▁⊔当地⊔媒体⊔报道⊔,⊔ 一⊔辆⊔机场⊔消⊔防⊔车 ⊔在⊔响⊔应⊔火⊔警⊔时 ⊔翻⊔了⊔车⊔。	30	⊔ال▁⊔و⊔سائل▁⊔أع⊔ل⊔□ت▁ ▁⊔عن▁⊔□مح⊔ل⊔ية▁⊔إعلام س⊔يارا□▁⊔إ⊔ح⊔دى⊔□قلاب توجه▁⊔أ⊔ثناء▁⊔الإ⊔طف⊔اء▁⊔ ⊔□حر⊔يق▁⊔لإ⊔طف⊔اء▁⊔ها⊔ .

Table 9: Tokenization examples obtained from the dev set in Flores+ (2/3). The '⊔' and '▁' symbols indicate the token delimiter and space character of SentencePiece, respectively.

Tokenizer/ vocabulary	Hindi #Tokens	Hindi Sample	Russian #Tokens	Russian Sample
M2M-100	28	▁स्थानीय⊔▁मीडिया⊔▁ने⊔▁बताया⊔▁है⊔▁कि⊔▁कार⊔्र⊔वाई⊔▁करने⊔▁के⊔▁दौरान⊔▁एयर⊔पोर्ट⊔▁का⊔▁अ⊔ग्⊔नि⊔शा⊔मक⊔▁वा⊔हन⊔▁लु⊔ढ़⊔क⊔▁गया⊔▁□ा⊔.	26	▁Ме⊔ст⊔ные⊔▁СМИ⊔▁сооб⊔ща⊔ют⊔,⊔▁что⊔▁в⊔▁аэ⊔ропо⊔рту⊔▁по⊔▁пути⊔▁на⊔▁вы⊔зов⊔▁пере⊔вер⊔ну⊔лась⊔▁пожар⊔ная⊔▁машина⊔.
mBART/XLM-R	22	▁स्थानीय⊔▁मीडिया⊔▁ने⊔▁बताया⊔▁है⊔▁कि⊔▁कार्रवाई⊔▁करने⊔▁के⊔▁दौरान⊔▁एयरपोर्ट⊔▁का⊔▁अग्नि⊔शा⊔मक⊔▁वाहन⊔▁लु⊔ढ़⊔क⊔▁गया⊔▁□ा⊔.	22	▁Мес⊔т⊔ные⊔▁СМИ⊔▁со⊔обща⊔ют⊔,⊔▁что⊔▁в⊔▁аэропорт⊔у⊔▁по⊔▁пути⊔▁на⊔▁вызов⊔▁пере⊔вернулась⊔▁пожар⊔ная⊔▁машина⊔.
NLLB-200	23	▁स्थानीय⊔▁मीडिया⊔▁ने⊔▁बताया⊔▁है⊔▁कि⊔▁कार्रवाई⊔▁करने⊔▁के⊔▁दौरान⊔▁एयर⊔पोर्ट⊔▁का⊔▁अग्नि⊔शा⊔मक⊔▁वाहन⊔▁लु⊔ढ़⊔क⊔▁गया⊔▁□ा⊔.	27	▁Ме⊔ст⊔ные⊔▁С⊔М⊔И⊔▁сооб⊔ща⊔ют⊔,⊔▁что⊔▁в⊔▁аэро⊔порту⊔▁по⊔▁пути⊔▁на⊔▁выз⊔ов⊔▁пере⊔вер⊔ну⊔лась⊔▁пожар⊔ная⊔▁машина⊔.
mT5	36	▁स्थान⊔ीय⊔▁मी⊔डिया⊔▁ने⊔▁बता⊔या⊔▁है⊔▁कि⊔▁कार्⊔र⊔वाई⊔▁कर⊔ने⊔▁के⊔▁द⊔◌⊔रान⊔▁ए⊔यर⊔पोर्ट⊔▁का⊔▁अ⊔ग्नि⊔शा⊔मक⊔▁⊔वाह⊔न⊔▁लु⊔ढ़⊔क⊔▁ग⊔या⊔▁□ा⊔.	25	▁⊔Мест⊔ные⊔▁С⊔МИ⊔▁сообщ⊔ают⊔,⊔▁что⊔▁в⊔▁аэро⊔порту⊔▁по⊔▁⊔пути⊔▁на⊔▁выз⊔ов⊔▁пере⊔верну⊔лась⊔▁пожар⊔ная⊔▁машина⊔.
LlaMa2	102	▁⊔स⊔्⊔□⊔ा⊔न⊔ी⊔य⊔▁⊔म⊔ी⊔ड⊔ि⊔य⊔ा⊔▁⊔न⊔े⊔▁⊔ब⊔त⊔ा⊔य⊔▁⊔ह⊔ै⊔▁⊔क⊔ि⊔▁⊔क⊔ा⊔र⊔्⊔र⊔व⊔ा⊔<0xE0>⊔<0xA4>⊔<0x88>⊔▁⊔क⊔र⊔न⊔े⊔▁⊔क⊔े⊔▁⊔द⊔<0xE0>⊔<0xA5>⊔<0x8C>⊔र⊔ा⊔न⊔▁⊔<0xE0>⊔<0xA4>⊔<0x8F>⊔य⊔र⊔प⊔ो⊔र⊔्⊔ट⊔▁⊔क⊔ा⊔▁⊔अ⊔ग⊔्⊔न⊔ि⊔श⊔ा⊔म⊔क⊔▁⊔व⊔ा⊔ह⊔न⊔▁⊔ल⊔ु⊔<0xE0>⊔<0xA4>⊔<0xA2>⊔़⊔क⊔▁⊔ग⊔य⊔ा⊔▁⊔□⊔ा⊔.	35	▁Ме⊔ст⊔ные⊔▁С⊔М⊔И⊔▁сооб⊔ща⊔ют⊔,⊔▁что⊔▁в⊔▁а⊔э⊔ро⊔пор⊔ту⊔▁по⊔▁пу⊔ти⊔▁на⊔▁вы⊔зов⊔▁пере⊔вер⊔ну⊔лась⊔▁по⊔ж⊔ар⊔ная⊔▁ма⊔ши⊔на⊔.
250K_S+B	22	▁स्थानीय⊔▁मीडिया⊔▁ने⊔▁बताया⊔▁है⊔▁कि⊔▁कार्रवाई⊔▁करने⊔▁के⊔▁दौरान⊔▁एयरपोर्ट⊔▁का⊔▁अग्नि⊔शा⊔मक⊔▁वाहन⊔▁लु⊔ढ़⊔क⊔▁गया⊔▁□ा⊔.	23	▁Мес⊔т⊔ные⊔▁СМИ⊔▁сообща⊔ют⊔,⊔▁что⊔▁в⊔▁аэропорт⊔у⊔▁по⊔▁пути⊔▁на⊔▁вы⊔зов⊔▁пере⊔верну⊔лась⊔▁пожар⊔ная⊔▁машина⊔.
64K_S+B	29	▁स्थानीय⊔▁मीडिया⊔▁ने⊔▁बताया⊔▁है⊔▁कि⊔▁कार⊔्र⊔वाई⊔▁करने⊔▁के⊔▁दौरान⊔▁ए⊔यर⊔पोर्ट⊔▁का⊔▁अ⊔ग्नि⊔शा⊔मक⊔▁वा⊔हन⊔▁⊔लु⊔ढ़⊔क⊔▁गया⊔▁□ा⊔.	30	▁Ме⊔ст⊔ные⊔▁С⊔МИ⊔▁сообщ⊔а⊔ют⊔,⊔▁что⊔▁в⊔▁а⊔эр⊔опо⊔рту⊔▁по⊔▁пут⊔и⊔▁на⊔▁вы⊔зов⊔▁пере⊔верну⊔лась⊔▁по⊔жар⊔ная⊔▁машин⊔а⊔.

Table 10: Tokenization examples obtained from the dev set in Flores+ (3/3). The '⊔' and '▁' symbols indicate the token delimiter and space character of SentencePiece, respectively.