Byte-based Multilingual NMT for Endangered Languages

Mengjiao Zhang
Stevens Institute of Technology
mzhang49@stevens.edu

Jia Xu
Stevens Institute of Technology
jxu70@stevens.edu

Abstract

Multilingual neural machine translation (MNMT) jointly trains a shared model for translation with multiple language pairs. However, traditional subword-based MNMT approaches suffer from out-of-vocabulary (OOV) issues and representation bottleneck, which often degrades translation performance on certain language pairs. While byte tokenization is used to tackle the OOV problems in neural machine translation (NMT), until now its capability has not been validated in MNMT. Additionally, existing work has not studied how byte encoding can benefit endangered language translation to our knowledge. We propose a byte-based multilingual neural machine translation system (BMNMT) to alleviate the representation bottleneck and improve translation performance in endangered languages. Furthermore, we design a random byte mapping method with an ensemble prediction to enhance our model robustness. Experimental results show that our BMNMT consistently and significantly outperforms subword/word-based baselines on twelve language pairs up to +18.5 BLEU points, an 840% relative improvement.

1 Introduction

Neural Machine Translation (NMT) has achieved great success and dominates recent research on translation tasks in both academic and industry studies (Wu et al., 2016; Stahlberg, 2020; Chen et al., 2018). In particular, multilingual neural machine translation (MNMT) has been shown to benefit low-resource language translation by jointly training MNMT models with high-resource languages (Johnson et al., 2017). However, most existing NMT/MNMT models are based on word or subword tokenization, which has three main problems. First, language-specific tokenizers, such as BPE (Shaham and Levy, 2021), may introduce inaccurate segmentations (Banerjee and Bhattacharya, 2018). Second, out-of-vocabulary (OOV) words/subwords are still unavoidable and hurt translation performance. Third, some languages show a decrease in translation quality with multilingual training due to specific characteristics of the language variety, a problem known as the representation bottleneck (Dabre et al., 2020; Zoph and Knight, 2016). It essentially limits the improvement of transfer learning from high-resource NMT, like Chinese-English (Gu et al., 2018) to the low-resource NMT, like Aymara-Spanish.

Recently, byte-based NMT models show comparable performance to word/subword-based models (Shaham and Levy, 2021; Wang et al., 2020). Because modern byte encoding systems such as UTF-8 have only 256-byte entries in total, i.e., `0x00` to `0xff`, the unified byte tokenization in all languages obviates the traditional preprocessing of language-specific subword tokenization and restricts the vocabulary to a fixed and small one. Therefore, the 256-byte-sized UTF-8 vocabulary avoids the OOV issues.

Despite the advantages of byte tokenization, the byte encoding has not been investigated in multilingual NMT yet, to the best of our knowledge. In this paper, we show that byte tokenization can be naturally applied to MNMT systems with great advantages. Given the unified encoding of byte tokenization in all languages, byte encoding is able to address the representation bottleneck problem in MNMT systems effectively. For example, Figure 1 shows the overlaps of subword and byte vocabularies among multiple languages. We notice with the growth of the language number, there is almost no overlap of word vocabularies, while the byte vocabulary still has a large overlap. This large overlap of byte vocabulary enables enhanced knowledge sharing among different languages and can help the model learn more generalized representations for these languages. Taking the low resource as a common property for most endangered languages,

Proceedings of the 29th International Conference on Computational Linguistics, pages 4407–4417
October 12–17, 2022.

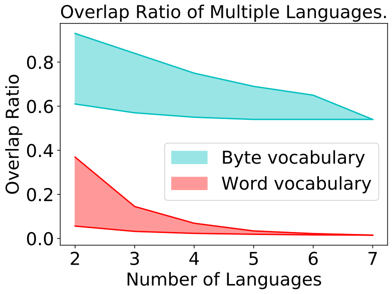


Figure 1: The vocabulary overlap ratio of byte and subword tokens. The area upper/lower bound denotes the highest/lowest ratio among all language combinations, given a language number.

we think byte-based MNMT is particularly helpful for endangered language translation. Therefore, we aim to incorporate byte tokenization in MNMT to alleviate the representation bottleneck problem in multilingual translation.

Besides incorporating byte encoding into MNMT, we aim to further investigate the generalizability of byte encoding. We observed that byte mapping can be arbitrary. For example, the characters "a"-"z" are represented with bytes 97-122 using UTF-8. However, we conjecture "a"-"z" can be any byte from 0 to 255. The byte representation does not need to be determined as a single encoding mapping. However, existing byte-based NMT systems do not consider such randomness of byte encoding. Moreover, we think language models should provide similar performance given different byte encoding methods to improve the generalizability and robustness. Therefore, we design a new encoding method that we call Random Byte Encoding by incorporating the random representation of bytes and reduce the variance of model outputs.

In this work, to address these challenges, we propose a **B**yte-based **M**ultilingual **N**eural **M**achine **T**ranslation framework (**BMNMT**). It simultaneously considers the byte randomness and the endangered languages in multilingual translation and works as follows. First, we design a novel MNMT framework that can take the byte encoding of sentences as inputs. Then, we incorporate the randomness of the byte encoding as discussed above by generating random byte mapping to replace the original byte ordering. We finally propose an ensemble prediction method by combining different encodings for reliable outputs.

Our BMNMT achieves amazing results in improving the low-resource and the endangered language translation that often does not have satisfactory results due to scarce resources and other linguistic characteristics (Levow et al., 2021; Ens et al., 2019; Liu et al., 2022). We demonstrate that our BMNMT consistently and significantly enhances the translation performance on all languages, including five high-resource languages, German, Arabic, Chinese, Farsi, Turkish to English, one low-resource language, Slovenian to English, and ten endangered languages, Asháninka, Aymara, Bribri, Guarani, Nahuatl, Otomí, Quechua, Rarámuri, Shipibo-Konibo, and Wixarika to Spanish. For example, the translation BLEU score [%] increases from 0 to 3.9 for the endangered language Shipibo-Konibo, and from 2.2 to 20.7 for the low-resource language Slovenian to English, i.e., +18 BLEU points. The contributions of this work are summarized as follows:

- We propose an effective byte-based MNMT framework to alleviate the representation bottleneck problem in word/subword-based multilingual translation, especially for endangered languages.
- We design a novel method of random byte encoding with ensemble prediction to enhance the generalizability and robustness of our byte-based MNMT model.
- We evaluate BMNMT on various training strategies. Extensive experiments validate the effectiveness, generalizability, and robustness of our model.

In the following context, we first outline the previous work in Section 2, then describe our method in Section 3, finally show our experimental results in Section 4 and conclude this work in Section 6.

2 Related Work

Multilingual Neural Machine Translation Word and subword-level tokenizations are widely used in natural language processing, including NMT/MNMT. Morishita et al. (Morishita et al., 2018) propose to incorporate hierarchical subword features to improve neural machine translation. Massively multilingual NMT models are proposed by Aharoni et al. (Aharoni et al., 2019) and Arivazhagan et al. (Arivazhagan et al., 2019). They are trained on massive language pairs and show a strong and positive impact on low-resource languages. However, these models tend to have representation bottlenecks (Dabre et al., 2020), due to large vocabulary size and large diversity of training languages. Two MNMT systems (Tan et al., 2019; Pan et al., 2021) are proposed to solve this problem by modifying the model architectures, adding special constraints on training, or designing more complicated preprocessing methods. Pan et al. (Pan et al., 2021) adopt the contrastive learning scheme in many-to-many MNMT. Tan et al. (Tan et al., 2019) propose a distillation based approach to boost the accuracy of MNMT systems. However, these word/subword-based models still need complex preprocessing steps such as data augmentation or special model architecture design.

Byte tokenizaiton Recently, byte tokenization methods are proposed to address the OOV problems in word/subword-based models. Ruiz et al. (Ruiz Costa-Jussà et al., 2017) compare character-based and byte-based NMT systems and show that byte-based systems converge faster. Wang et al. (Wang et al., 2020) combine subwords tokenization with byte encoding and propose a byte-level BPE (BBPE). Shaham and Levy (Shaham and Levy, 2021) propose embeddingless byte-to-byte machine translation by replacing the token embedding layer in subword-based models with one-hot encoding for bytes. However, among these models, byte-level MNMT is still not studied, and the randomness of byte tokenization as we discussed above is not investigated.

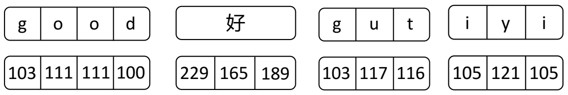
Therefore, different from the previous work, we mainly focus on byte-based MNMT, while simultaneously considering the randomness of bytes and endangered languages.

3 Methods

3.1 Preliminary of Byte Representation

Any writing system can be encoded with a byte sequence (Needleman, 2000; Shaham and Levy, 2021), using pre-defined byte encoding methods, such as UTF-8 for almost all languages, GBK for simplified Chinese, and eucJP for Japanese.

Formally, we use a mapping function $f : \mathcal{C} \rightarrow \mathcal{B}^n$ to denote the mapping from characters in a raw sentence to bytes. Here, $\mathcal{C}$ is the character domain for all languages, $\mathcal{B} = (0, 1, \ldots, 255)$ is the byte domain, and n is the maximum byte number that a character maps into. Also, we define f^{-1} on a byte sequence to convert it back to the text. In this paper, we use UTF-8 as the mapping function, because it is a general encoding method and contains almost all characters in existing languages. Figure 2 shows four different languages represented in bytes, including English (en), Chinese (zh), German (de), and Turkish (tr). A character in each language is mapped into bytes. Particularly, characters of some languages such as Chinese are mapped into multiple bytes.



(a) en (b) zh (c) de (d) tr

Figure 2: Byte representation of four languages.

3.2 Problem Definition

Here, we first describe the input and output of the multilingual translation task.

Definition 1 (Multilingual Domain)**.** *We use* $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N\}$ *to denote the source language domain and use* $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_M\}$ *to denote the target language domain.* $\mathcal{S}_i$ *or* $\mathcal{T}_j$ *represents a type of language. Note that* $\mathcal{S}$ *and* $\mathcal{T}$ *can have intersection.*

Definition 2 (Multilingual Sentence Pair)**.** *Given the source domain* $\mathcal{S}$ *and the target domain* $\mathcal{T}$*, we define the multilingual sentence pair set* $\mathcal{L} = \{(s_i, s_j) \mid s_i \in \mathcal{S}_i \text{ and } s_j \in \mathcal{T}_j \text{ and } \mathcal{S}_i \neq \mathcal{T}_j\}$*. Here, the* s_i *and* s_j *denote two parallel sentences from different languages.*

Based on the above description, we formally define the multilingual translation task as follow:

Definition 3 (Multilingual Translation)**.** *Given the multilingual sentence pair* $\mathcal{L}$ *as the training set and a translation model* $\mathcal{M}$ *with parameters* $\boldsymbol{\theta}$*, we aim to find the optimized parameters* $\hat{\boldsymbol{\theta}}$ *of* $\mathcal{M}$ *to minimize the following objective function:*

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{(s_i, s_j) \in \mathcal{L}} -p(s_j) \log p(\hat{s}_j | s_i; \boldsymbol{\theta}). \quad (1)$$

Here, $\hat{s}_j$ is the predicted sentence of the target sentence s_j. To generate each token $\hat{x}_j^t$ in $\hat{s}_j$, we use s_i and the previous tokens $s_j^{<t}$ of s_j to calculate a probability $p(\hat{x}_j^t) = \mathcal{M}(s_i, s_j^{<t}; \boldsymbol{\theta})$. Assume s_j consists of m tokens (byte or subword): $(x_j^1, x_j^2, \ldots, x_j^m)$, the conditional probabil-

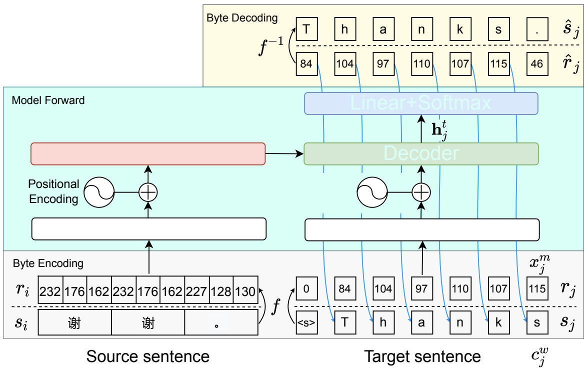


Figure 3: The overview of the proposed byte-based multilingual translation model BMNMT.

ity $p(\hat{s}_j|s_i;\boldsymbol{\theta})$ is defined as:

$$p(\hat{s}_j|s_i;\boldsymbol{\theta}) = \prod_{t=1}^{m} p(\hat{x}_j^t) = \prod_{t=1}^{m} \mathcal{M}(s_i, s_j^{<t}; \boldsymbol{\theta}). \tag{2}$$

3.3 Multilingual translation with byte-level tokenization

Suppose we are given a sentence pair $(s_i, s_j) \in \mathcal{L}$ and $s_i = (c_i^1, \ldots, c_i^k), s_j = (c_j^1, \ldots, c_j^w)$, respectively. c is a character of s_i or s_j. k and w denote the character number of the raw source sentence s_i and target sentence s_j, respectively. Our proposed multilingual translation framework BMNMT contains three main parts: byte encoding, model forward, and byte decoding. The model overview of BMNMT is shown in Figure 3.

Byte encoding We first use the byte mapping function f to encode the raw sentence pairs into byte sequences. Take s_j in Figure 3 as an example. We first map w characters in each raw sentence into a new sequence r_j of m byte tokens:

$$\begin{aligned} r_j &= (f(c_j^1), f(c_j^2), \ldots, f(c_j^w)) \\ &= (x_j^1, x_j^2, \ldots, x_j^m). \end{aligned} \tag{3}$$

It is worth noting that we also consider the punctuation and the space symbol in the raw text as characters and encode them into bytes.

Model forward Our byte-level multilingual translation model is based on the state-of-the-art Transformer architecture (Vaswani et al., 2017), which includes an encoder Enc and a decoder Dec. After getting the byte tokens of sentence pairs, we convert r_i and r_j to one-hot vectors $\mathbf{r}_i$ and $\mathbf{r}_j$. Then, the encoder encodes the source sequence into hidden representations, and the decoder outputs logits $\mathbf{h}_j^t \in \mathbb{R}^d$ for each generated token $\hat{x}_j^t$ of the target sentence. Here, d is the dimension of the decoder output. Finally, $\mathcal{M}$ calculates a probability $p(\hat{x}_j^t)$ with a fully-connected (FC) layer with Softmax:

$$\mathbf{h}_j^t = \texttt{Dec}(\texttt{Enc}(\mathbf{r}_i), \mathbf{r}_j^{<t}) \in \mathbb{R}^d, \tag{4}$$

$$p(\hat{x}_j^t) = \text{Softmax}(\mathbf{W}\mathbf{h}_j^t) \in \mathbb{R}^{256}. \tag{5}$$

Here, $\mathbf{W} \in \mathbb{R}^{256\times d}$ is the weight in the FC layer to project the output space of the decoder into the byte space.

Byte decoding After getting the probability distribution $p(\hat{x}_j^t)$ of the current output token $\hat{x}_j^t$, we then use Beam Search (BS) to sample the target byte token $\hat{x}_j^t$. Finally, after generating the entire byte sequence, we use the inverse mapping f^{-1} to retrieve the real generated text sentence $\hat{s}_j$:

$$\hat{x}_j^t = \texttt{BS}(p(\hat{x}_j^t)) \tag{6}$$

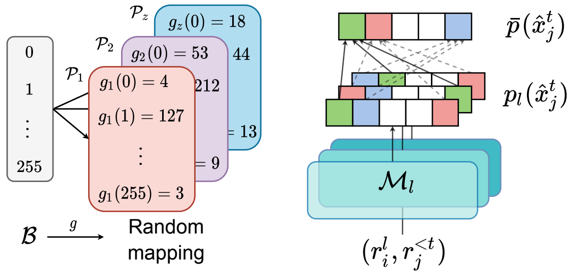
$$\hat{s}_j = f^{-1}(\hat{x}_j^1, \hat{x}_j^2, \ldots, \hat{x}_j^m) \tag{7}$$

Note that, in the inference process, we autoregressivelly output the target tokens using the previous generated tokens $\hat{s}_j^{<t}$ instead of $s_j^{<t}$, i.e., $p(\hat{x}_j^t) = \mathcal{M}(s_i, \hat{s}_j^{<t}; \boldsymbol{\theta})$.

3.4 Random byte encoding and ensemble prediction

As discussed in the previous section, we use a one-hot vector to represent a byte token. For example, in Figure 3, the byte representation of character “h” is 104 under UTF-8 encoding. In the one-hot vector, the entry in 104 is one and the others are 0. However, we conjecture UTF-8 is just one of the mapping functions for bytes. The language model should not be limited by a single byte encoding method, because a single encoding method can bring possible bias in model representation. Rare resources of data and other characteristics make it harder for endangered languages on translation task. Therefore, we propose an additional random byte encoding method besides the basic byte tokenization by generating multiple random byte mappings. Then we design an ensemble prediction by training multiple models $\mathcal{M}$s with different byte mappings and output an average probability among all $\mathcal{M}$s to enhance the robustness of the translation model.

Random byte mapping To generate multiple random mappings, we define z permutation functions $g_1, \ldots, g_z$ by shuffling the original bytes $\mathcal{B} = (0, 1, \ldots, 255)$ to $\mathcal{P}_1, \ldots, \mathcal{P}_z$. Here, g_l is



(a) Random byte mapping (b) Ensemble prediction

Figure 4: Random byte mapping and ensemble prediction. (a) Random byte mapping with z different byte permutations of UTF-8. (b) Ensemble prediction by producing z BMNMT for each byte permutation.

a one-to-one mapping from $\mathcal{B}$ to $\mathcal{P}_l$. Figure 4(a) shows one example of the random byte mapping. $g_1(0) = 4$ maps byte 0 in $\mathcal{B}$ to byte 4 in $\mathcal{P}_1$.

Ensemble prediction In the previous section, we directly use byte encoding from UTF-8. By introducing random byte encoding, we first update Equation (3) using the byte mapping g_l:

$$r_j^l = \left(g_l\left(x_j^1\right), g_l\left(x_j^2\right) \ldots, g_l\left(x_j^m\right)\right) \qquad (8)$$

Then, we have z byte-level input sentence pairs $(r_i^1, r_j^1), (r_i^2, r_j^2), \ldots, (r_i^z, r_j^z)$ for each (s_i, s_j). For l-th random byte encoding, we adopt an individual multiliangual Transformer $\mathcal{M}_l$ to take the l-th byte-level language pair under mapping $g_l(\cdot)$ as the the input. As a result, we calculate z probability distributions $p_1(\hat{x}_j^t), p_2(\hat{x}_j^t), \ldots, p_z(\hat{x}_j^t)$ from Equation (5). Next, we ensemble these z distributions. $p_l(\hat{x}_j^t)$ is a vector of 256 entries. For each entry $v \in \mathcal{B}$, we find the corresponding entry $g_l(v)$ in probability distributions $p_l(\hat{x}_j^t)$ of $\mathcal{P}_l$ and calculate an average probability from $\mathcal{P}_1$ to $\mathcal{P}_z$:

$$\bar{p}^v(\hat{x}_j^t) = \frac{1}{z}\sum_{l=1}^{z} p_l^{g_l(v)}(\hat{x}_j^t). \qquad (9)$$

We still take Figure 4(a) as an example. When calculating the probability of byte 0 in $\mathcal{B}$, we find byte 4 in $\mathcal{P}_1$, byte 53 in $\mathcal{P}_2, \ldots$, and byte 18 in $\mathcal{P}_z$. Finally, we use $\bar{p}(\hat{x}_j^t)$ to execute beam search instead of $p(\hat{x}_j^t)$ in Equation (6). The ensemble prediction modules are demonstrated in Figure 4(b).

4 Experimental Settings

4.1 Dataset

In our experiments, we use the English-centric IWSLT14 dataset (Cettolo et al., 2014) and the IndCorpus dataset (Chen et al., 2021).

Specifically, in IWSLT14, we select five high-resource language pairs, i.e., Chinese, Arabic, German, Farsi, Turkish to English, and one low-resource language pair, i.e., Slovenian to English. When preprocessing IWSLT14, we remove sentences longer than 800 bytes in the training set, following the settings in (Shaham and Levy, 2021). In total, about 5% samples are removed.

The IndCorpus contains ten endangered languages. We translate each language to Spanish. Because the entire test set of IndCorpus is not publicly available, we adopt the Dev set as the test data. The statistics of IWSLT14 and IndCorpus can be found in Appendix A. Note that, although some endangered languages in IndCorpus have a large number of training samples, we still regard them as endangered languages because they have the characteristics of endangered languages.

4.2 Baselines and Model Settings

To study the representation bottleneck when upgrading from monolingual translation to multilingual translation, we first include a byte-based monolingual model. Next, to validate the effectiveness of byte-based MNMT, we also incorporate two subword-based monolingual and multilingual models. Specifically, we select the following model schemes as baselines with different model architectures on two datasets:

- *B-N* (Shaham and Levy, 2021). To valid the ability of our byte-based MNMT model BMNMT in alleviating the representation bottleneck, we select a byte-based NMT model (B-N) for monolingual translation.

- *W-N* (Mager et al., 2021). To compare the translation performance between our model and subword-based model, we select a subword-based NMT model (W-N) for monolingual translation.

- *W-M*. We implement a subword-based MNMT model (W-M) for multilingual translation based on W-N to further evaluate the effectiveness of our byte-based model on multilingual.

The framework of all models including our model are Transformers with the same architecture on each dataset. For the model scheme in IWSLT14, we follow the architecture in (Shaham and Levy, 2021). For IndCorpus, we follow the architecture in (Mager et al., 2021). For both tokenizations, all language pairs share the same model and the same dictionary. For the subword-based models, the embedding layers are shared among source and target languages. For byte-based models, we remove the embedding layers following (Shaham and Levy, 2021). The detailed model settings on two datasets including architecture and environment are listed in Appendix B. The source code is available at the Github repo: `https://github.com/MengjiaoZhang/Byte-based-multilingual-NMT`.

4.3 Hyper-parameters

Following (Shaham and Levy, 2021) and (Mager et al., 2021), for subword tokenization with BPE, we use 10,000 merging steps. The dropout rates of models in IWSLT14 and IndCorpus are 0.2 and 0.4, respectively. Due to the limited training data in IndCorpus 2021, we also set attention dropout as 0.2 and ReLU dropout as 0.2 to avoid over-fitting in IndCorpus. The optimizer is Adam (Kingma and Ba, 2015) with the inverse square root learning rate scheduler. We set the warm-up steps as 4,000 and the minimum learning rate is 10^{-7}. The training epoch for IWSLT14 is 200 with early stop when observing no decrease of validation loss within 5 consecutive epochs. The epoch number for IndCorpus is 50 without early stop because we use the dev set in IndCorpus as the test data.

4.4 Evaluation Metrics

To evaluate the translation results of all models, we use the commonly adopted BLEU scores on both datasets and use an additional metric ChrF (Popović, 2015) to evaluate endangered languages in IndCorpus. Here, ChrF denotes the character n-gram F-score. We adopt ChrF because not all endangered languages in IndCorpus have a tokenization standard (Mager et al., 2021).

4.5 Training strategies

To analyze the causes of the representation bottleneck and explore model generalizability while studying the effectiveness of the basic BMNMT framework and ensemble prediction, we consider multiple scenarios related to languages resource.

Case 1. Jointly **T**rain BMNMT with both **H**igh- and **L**ow-resource language pairs without finetuning and ensemble prediction (T-HL).

Case 2. Jointly **T**rain BMNMT only with the **H**igh resource language pairs without finetuning and ensemble prediction (T-H).

	Byte (BLEU)		Subword (BLEU)	
	B-N	BMNMT (Ours)	W-N	W-M
ar-en	30.8	30.4 (**-0.4**)	30.5	28.8 (-1.7)
de-en	34.4	34.2 (**-0.2**)	34.1	33.1 (-1.0)
fa-en	22.7	24.2 (**+1.5**)	21.6	23.0 (+1.4)
tr-en	22.8	22.5 (**-0.3**)	22.2	21.9 (-0.3)
zh-en	15.8	15.8 (**+0.0**)	15.9	15.8 (-0.1)
sl-en	2.2	20.7 (**+18.5**)	8.9	20.2 (+11.3)
Avg.	21.4	24.6 (**+3.2**)	22.2	23.8 (+1.6)

Table 1: Representation bottleneck analysis on Case 1 (T-HL) using the IWSLT14 dataset. Values in "()" represents the BLEU score difference between multilingual and monolingual models. Here, Slovenian (sl) is a relatively low-resource language.

Case 3. Jointly **T**rain BMNMT on the **E**ndangered language pairs without finetuning and ensemble prediction (T-E).

Case 4. **F**inetune BMNMT with **E**ndangered languages on pretrained BMNMT in Case 2 (F-E2).

Case 5. Jointly **T**rain BMNMT on **E**ndangered languages without finetuning but with ensemble prediction (T-E+P).

For the first four cases without ensemble, we use the original UTF-8 encoding. For the last case, we adopt the proposed random byte encoding method.

5 Experimental Results

In the main paper, we report the BLEU scores of all cases. The ChrF for endangered languages in Cases 3-5 can be found in Appendix C.

5.1 Representation Bottleneck Analysis on Cases 1 (T-HL) and 2 (T-H)

To validate the effectiveness of our model in addressing the representation bottleneck in multilingual translation with both high and low resource languages, we adopt the training strategies of Cases 1 (T-HL) and 2 (T-H). First, we run the subword-based translation baselines on monolingual (W-N) and multilingual (W-M) languages and calculate the difference of BLEU scores between them. Then, we also calculate such difference between BMNMT and B-N to evaluate whether BMNMT can alleviate the representation bottleneck in MNMT.

Table 1 shows the results of Case 1 (T-HL) on IWSLT14 dataset. In this experiment, the high-resource language pairs are Arabic (ar), German

	Byte (BLEU)		Subword (BLEU)	
	B-N	BMNMT (Ours)	W-N	W-M
ar-en	30.8	31.5 (**+0.7**)	30.5	29.6 (-0.9)
de-en	34.4	35.2 (**+0.8**)	34.1	33.6 (-0.5)
fa-en	22.7	24.9 (**+2.2**)	21.6	23.7 (+2.1)
tr-en	22.8	23.8 (**+1.0**)	22.2	22.4 (+0.5)
zh-en	15.8	16.9 (**+1.1**)	15.9	16.2 (+0.4)
Avg.	25.3	26.5 (**+1.2**)	24.9	25.1 (+0.2)

Table 2: Representation bottleneck analysis on Case 2 (T-H) with all high-resource languages in IWSLT14.

	Byte (BLEU)		Subword (BLEU)	
	B-N	BMNMT (Ours)	W-N	W-M
quy-es	2.4	3.5 (**+0.9**)	3.3	2.7 (-0.6)
gn-es	3.6	4.4 (**+0.8**)	2.1	3.0 (+0.9)
nah-es	0.2	2.6 (**+2.4**)	0.8	2.1 (+1.3)
shp-es	0.0	3.9 (**+3.9**)	0.3	2.8 (+2.5)
Avg.	1.6	3.6 (**+2.0**)	1.6	2.7 (+1.1)

Table 3: Representation bottleneck analysis on Case 3 (T-E) with endangered languages in IndCorpus. Quechua (quy) is a relatively high-resource endangered language.

(de), Farsi (fa), Turkish (tr), and Chinese (zh) to English (en), while the low-resource language pair is Slovenian (sl) to English. We first notice byte-level models achieve the best performance on almost all language pairs. It proves the capability of byte tokenization in the translation task. Furthermore, as a low-resource language, the translation from Slovenian to English gains the largest benefit from BMNMT, and BMNMT has the best average BLEU score. Therefore, we can conclude that the application of byte tokenization to MNMT is able to enhance the knowledge sharing among multiple languages.

In addition, we notice that subword-based multilingual translation (W-M) suffers from representation bottleneck, because the performance of some high-resource language pairs ar-en, de-en, tr-en, and zh-en decrease compared with monolingual translation (W-N). However, our proposed byte-based multilingual translation shows a much smaller decrease than word-based models. It further proves the ability of BMNMT to alleviate the representation bottleneck problem in MNMT.

Table 2 shows the results of Case 2 (T-H). Here, to further study the influence of language resource on the representation bottleneck, we remove the low-resource language Slovenian (sl) in Case 1 (T-HL) and re-train BMNMT and other baselines. We notice that after removing the language Slovenian, all MNMT models based on byte and subword gain a higher performance in the average BLEU score. However, subword-based W-M still cannot avoid the representation bottleneck, while our BMNMT achieves improvement on all pairs against the B-N model. More importantly, BMNMT has the best performance on all language pairs.

In summary, when training MNMT with both high and low-resource languages, the low-resource languages are a main reason for the representation bottleneck. Moreover, compared to subword-based models, our proposed BMNMT can effectively alleviate this problem.

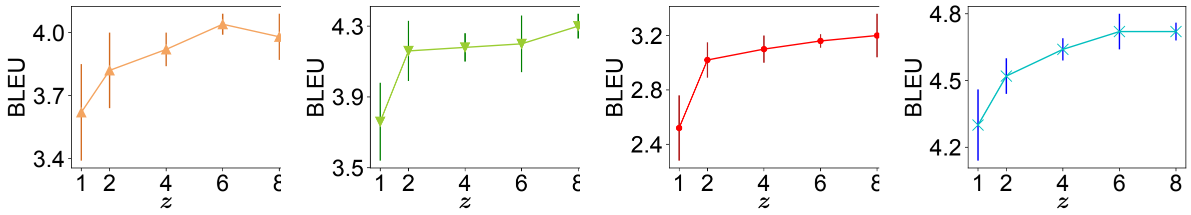
5.2 Representation Bottleneck Analysis on Endangered Languages: Case 3 (T-E)

To evaluate the ability of BMNMT to address the representation bottleneck in endangered languages, we adopt the training strategy of Case 3 (T-E). Here, we still report the BLUE score differences between monolingual and multilingual models.

Table 3 demonstrates the results of Case 3. Following the setting in Case 1, we select four language pairs Quechua (quy), Guarani (gn), Nahuatl (nah), and Shipibo-Konibo (shp) to Spanish (es) that have relatively large training sizes in IndCorpus to avoid the bias in languages of extremely low resource. Among these pairs, quy has the most training data that are comparable to high-resource languages in IWSLT14, while the others can be regarded as low-resource languages. We notice the BLEU score of quy is much lower than languages in IWSLT14. We infer it is because of the characteristics in endangered languages. However, different to the results in Case 1, the BLEU score of byte-based BMNMT on quy-es does not decrease even with multilingual training. To summarize, we think our BMNMT can help overcome the representation bottleneck of rare resource in endangered languages and bring enhancement in translation.

5.3 Generalizability Analysis: Case 4 (F-E2)

Another property we want to analyze for multilingual translation is the model generalizability. Therefore, we adopt Case 4 (F-E2) to generalize the pretrained model on high-resource languages to endangered languages. As the case indicates, we use the pretrained BMNMT and W-M in Case 2, which are trained on all high-resource languages and individually finetuned on all endangered lan-



(a) quy-es (b) shp-es (c) nah-es (d) gn-es

Figure 5: Robustness analysis on Case 5 (T-E+P): Average BLEU scores of ensemble prediction with different ensemble number z.

	Byte (BLEU)		Subword (BLEU)	
	B-N	BMNMT (Ours)	W-N	W-M
gn-es	4.5	5.9 **(+1.4)**	3.6	4.0 (+0.4)
nah-es	3.4	4.5 **(+1.1)**	2.8	2.0 (-0.8)
quy-es	7.0	7.9 **(+0.9)**	5.4	5.9 (+0.4)
shp-es	0.6	2.7 **(+2.1)**	1.0	1.1 (+0.1)
aym-es	3.8	5.1 **(+1.3)**	2.8	2.7 (-0.1)
cni-es	0.3	1.8 **(+1.5)**	0.6	0.6 (+0.0)
bzd-es	0.7	2.5 **(+1.7)**	0.9	0.9 (+0.0)
oto-es	0.4	1.4 **(+1.0)**	0.4	0.4 (+0.0)
tar-es	0.2	0.7 **(+0.5)**	0.2	0.3 (+0.1)
hch-es	1.7	2.6 **(+0.9)**	1.1	0.9 (-0.2)
Avg.	2.3	3.5 **(+1.2)**	1.9	1.9 (+0.0)

Table 4: Generalizability analysis on Case 4 (F-E2) by finetuning with all endangered languages in IndCorpus based on pretrained models in Case 2.

guages. As monolingual models, B-N and W-N are pretrained on German to English and finetuned on endangered languages. It is worth noting that all endangered languages in finetuning do not occur in pretraining to validate the models' generalizability.

Table 4 shows the finetuning results on IndCorpus. We first notice that the models pretrained on multilingual languages perform better than monolingual models. It indicates that multilingual models have a stronger generalizability to endangered languages. Additionally, although the byte-based monolingual model B-N has worse performance than W-N on some language pairs, the BLEU scores of these pairs are largely increased by our BMNMT compared to W-M. It further validates the generalizability of BMNMT.

5.4 Robustness Analysis: Case 5 (T-E+P)

To evaluate the translation effectiveness and robustness of our proposed BMNMT on endangered languages, we generate multiple random byte mappings and adopt Case 5 (T-E+P) in the ensemble prediction experiment. Specifically, we first choose the number z of random byte mappings, i.e., ensemble number from $\{1, 2, 4, 6, 8\}$. For each z, we train BMNMT 5 times on IndCorpus with different permutations of bytes in UTF-8. The average and standard deviation of BLEU scores in 5 runs for each z are reported in Figure 5.

With the growth of z, the BLUE score shows an increasing trend. It proves that appropriate ensemble can improve the translation performance. Moreover, BMNMT without ensemble ($z = 1$) has the highest standard deviation. We infer byte-based translation with only one encoding method can brings noise to token representations. However, with multiple byte mapping and ensemble prediction, the translation becomes more stable. Therefore, we think that the random byte mapping and ensemble prediction can improve the robustness of byte-based translation.

In summary, based on all analysis for the representation bottleneck, generalizability, and robustness, the effectiveness of our proposed BMNMT is validated. We conclude that with the introduce of byte in MNMT and the ensemble prediction for byte mappings, the representation bottleneck can be alleviated, especially on endangered language.

6 Conclusion

Multilingual neural machine translation has been successful in enhancing low-resource languages because of knowledge sharing. To address the representation bottleneck in existing subword-based multilingual translation systems, we propose a byte-based MNMT model, BMNMT with the Transformer architecture. To improve the model generalizability and robustness, we further design an ensemble prediction method with random byte encoding. Our experimental results show that BMNMT can alleviate the representation bottleneck and has a stronger generalization ability compared with subword-based MNMT. Meanwhile, BMNMT with ensemble prediction improves the translation performance and robustness on endangered language translation tasks. Extending our byte-based method to large scale models and datasets is promising and can improve model performance, which will be our future work.

Acknowledgments

We appreciate the National Science Foundation (NSF) Award No. 1747728 and NSF CRAFT Award, Grant No. 22001 to fund this research. We are also thankful for the support of the Google Cloud Research Program. We thank Chang Lu for the comments.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. 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 3874–3884.

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

Tamali Banerjee and Pushpak Bhattacharya. 2018. Meaningless yet meaningful: Morphology grounded subword-level nmt. *NAACL HLT 2018*, page 55.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86.

Wei-Rui Chen, Muhammad Abdul-Mageed, Hasan Cavusoglu, et al. 2021. Indt5: A text-to-text transformer for 10 indigenous languages. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 265–271.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Jeff Ens, Mika Hämäläinen, Jack Rueter, Philippe Pasquier, et al. 2019. Morphosyntactic disambiguation in an endangered language setting. In *22nd Nordic Conference on Computational Linguistics (NoDaLiDa) Proceedings of the Conference*. Linköping University Electronic Press.

Jiatao Gu, Yong Wang, Yun Chen, Victor OK Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631.

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

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Gina-Anne Levow, Emily P Ahn, and Emily M Bender. 2021. Developing a shared task for speech processing on endangered languages. In *Proceedings of the Workshop on Computational Methods for Endangered Languages*, volume 1, pages 96–106.

Zoey Liu, Crystal Richardson, Richard Hatcher Jr, and Emily Prud'hommeaux. 2022. Not always about you: Prioritizing community needs when developing endangered language technology. *arXiv preprint arXiv:2204.05541*.

Manuel Mager, Arturo Oncevay, Abteen Ebrahimi, John Ortega, Annette Rios Gonzales, Angela Fan, Ximena Gutierrez-Vasques, Luis Chiruzzo, Gustavo Giménez-Lugo, Ricardo Ramos, et al. 2021. Findings of the americasnlp 2021 shared task on open machine translation for indigenous languages of the americas. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 202–217.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2018. Improving neural machine translation by incorporating hierarchical subword features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 618–629.

Mark Needleman. 2000. The unicode standard. *Serials review*, 26(2):51–54.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 244–258.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Marta Ruiz Costa-Jussà, Carlos Escolano Peinado, and José Adrián Rodríguez Fonollosa. 2017. Byte-based neural machine translation. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 154–158. Association for Computational Linguistics.

Uri Shaham and Omer Levy. 2021. Neural machine translation without embeddings. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 181–186.

Felix Stahlberg. 2020. Neural machine translation: A review. *Journal of Artificial Intelligence Research*, 69:343–418.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

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

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. *arXiv preprint arXiv:1601.00710*.

A Dataset Statistics

Tables 5 and 6 show the data statistics of IWSLT14 and endangered language dataset IndCorpus, respectively. IWSLT datasets contains English scripts of TED talks translated into other languages. All training samples in IWSLT are collected in 2014. The test samples are from TED talks from 2010 to 2012. For IndCorpus, there are ten indigenous languages from multiple countries of America.

B Model Architecture and Environment

In every Transformer model used in IWSLT14, we adopt 6 attention layers for both encoder for decoder. The number of attention heads is 4. For the subword-based models, they contain an embedding layer for tokens. The embedding dimension is 512, which is the same as the hidden dimension. The feed-forward layer is built upon the hidden layer to calculate the output digits. The dimension of feed-forward layers in the encoder and decoder is 1024. For byte-based models, they do not have the embedding layer.

ISO	Language	Train	Dev	Test
zh	Chinese	166,046	7,547	5,099
ar	Arabic	165,591	7,526	5,357
de	German	158,516	7,205	5,585
fa	Farsi	99,792	4,536	4,244
tr	Turkish	142,619	6,482	5,433
sl	Slovenian	15,859	720	2,555

Table 5: Languages from the IWSLT14 dataset with the ISO codes. The Train, Dev, and Test columns denote the number of sentence pairs of each language with English in the training, validation, and test set.

ISO	Language	Train	Dev
cni	Asháninka	3,883	883
aym	Aymara	6,531	996
bzd	Bribri	7,508	996
gn	Guarani	26,032	995
nah	Nahuatl	16,145	672
oto	Otomí	4,889	599
quy	Quechua	125,008	996
tar	Rarámuri	14,721	995
shp	Shipibo-Konibo	14,592	996
hch	Wixarika	8,966	994

Table 6: The languages featured in the IndCorpus, their ISO codes, and dataset statistics.

For the Transformer model in IndCorpus, we shrink the model size to avoid overfitting because the languages in IndCorpus only contain limited training samples. The encoder and decoder both have 5 attention layers with 2 heads. The remaining parts keep the same as the model architecture used in IWSLT14.

We use the transformer model implemented by faiseq[1]. All the program used in this work is implemented using Python 3.8, PyTorch 1.10.0, and CUDA 11.3. For the hardware environment, we run our program on a machine with Intel i9-10900KF CPU, 128G memory, and an NVIDIA GeForce RTX 3090 GPU.

C Additional Experimental Results on Endangered Languages using ChrF

Table 3 and Table 9 show the translation performance of endangered languages with ChrF in Cases 3 (T-E) and 4 (F-E2). Similar to the results in Ta-

[1] `https://github.com/facebookresearch/fairseq`

	IWSLT14	IndCorpus
Encoder layers	6	5
Decoder layers	6	5
Attention heads	4	2
Hidden dim d	512	512
Feed-forward dim	1024	1024

Table 7: Model architectures on the IWSLT14 and IndCorpus datasets.

	Byte (ChrF)		Subword (ChrF)	
	B-N	BMNMT (Ours)	W-N	W-M
quy-es	22.5	25.2 (**+2.7**)	21.5	20.3 (-0.8)
gn-es	21.8	23.4 (+1.6)	18.3	22.7 (**+4.4**)
nah-es	13.0	19.7 (**+6.7**)	15.0	19.6 (+4.6)
shp-es	12.0	27.0 (**+15.0**)	10.5	21.6 (+10.9)
Avg.	17.3	23.8 (**+5.5**)	16.7	21.2 (+4.5)

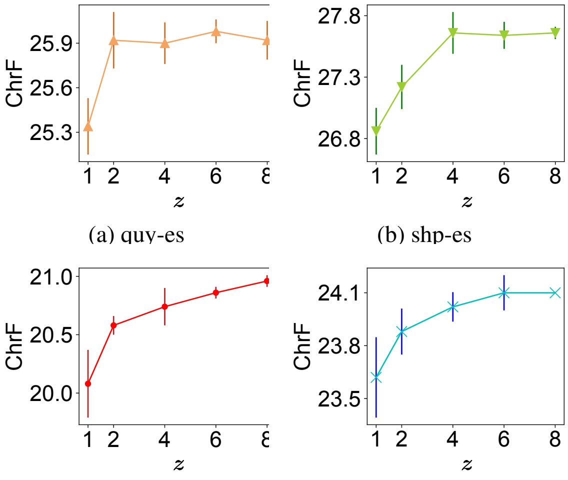
Table 8: Representation bottleneck analysis on Case 3 (T-E) with endangered languages. The evaluation metric in this table is ChrF.

ble 3 and Table 4 with BLEU scores, our proposed BMNMT has the best performance, and it can improve ChrF on all languages. Evaluation with these two metrics proves that the byte tokenization can alleviate the representation bottleneck even in endangered languages.

In Figure 6, we plot the ChrF scores of ensemble prediction. With the number of ensemble models increasing, the average translation performance shows the same trend while the standard deviation decrease. Both the evaluation on BLEU scores and ChrF show that the ensemble prediction with random byte mapping improve the translation performance and robustness in MNMT.

	Byte (ChrF)		Subword (ChrF)	
	B-N	BMNMT (Ours)	W-N	W-M
gn-es	25.7	27.1 (+**1.4**)	22.1	21.6 (-0.5)
nah-es	23.2	25.3 (**+2.1**)	18.1	17.7 (-0.4)
quy-es	32.1	32.9 (+0.8)	26.3	27.2 (**+0.9**)
shp-es	20.3	27.3 (**+7.0**)	16.2	17.0 (+0.8)
aym-es	22.9	25.8 (**+2.9**)	18.2	18.7 (+0.5)
cni-es	17.0	21.2 (**+4.2**)	14.1	13.7 (-0.4)
bzd-es	19.7	24.4 (**+4.7**)	14.3	15.0 (+0.7)
oto-es	17.9	19.4 (**+1.5**)	12.1	13.2 (+0.0)
tar-es	16.2	19.4 (**+3.2**)	15.1	14.7 (-0.4)
hch-es	20.3	23.6 (**+3.3**)	13.6	14.8 (+1.2)
Avg.	21.5	24.6 (**+3.1**)	17.0	17.4 (+0.4)

Table 9: Translation performance (ChrF) on IndCorpus. The results are finetune on the model trained in case 2.



(a) quy-es

(b) shp-es

(c) nah-es

(d) gn-es

Figure 6: Robustness analysis on Case 5 (T-E+P): Average ChrF of ensemble prediction with different ensemble number z.