Learning an Artificial Language for Knowledge-Sharing in Multilingual Translation

Danni Liu and **Jan Niehues**
Karlsruhe Institute of Technology
{danni.liu, jan.niehues}@kit.edu

Abstract

The cornerstone of multilingual neural translation is shared representations across languages. Given the theoretically infinite representation power of neural networks, semantically identical sentences are likely represented differently. While representing sentences in the *continuous* latent space ensures expressiveness, it introduces the risk of capturing of irrelevant features which hinders the learning of a common representation. In this work, we *discretize* the encoder output latent space of multilingual models by assigning encoder states to entries in a codebook, which in effect represents source sentences in a new artificial language. This discretization process not only offers a new way to interpret the otherwise black-box model representations, but, more importantly, gives potential for increasing robustness in unseen testing conditions. We validate our approach on large-scale experiments with realistic data volumes and domains. When tested in zero-shot conditions, our approach is competitive with two strong alternatives from the literature. We also use the learned artificial language to analyze model behavior, and discover that using a similar bridge language increases knowledge-sharing among the remaining languages.

1 Introduction

A promising potential of multilingual (Dong et al., 2015; Firat et al., 2016; Ha et al., 2016; Johnson et al., 2017) neural machine translation (NMT) is knowledge-sharing between languages. To enable knowledge-sharing, a prerequisite is the ability to capture common features of languages, especially between related ones. *Constructed languages* such as *Interlingua* and *Esperanto* are excellent examples of human-designed structures based on the commonalities of a wide range of related languages. For data-driven models, however, it is difficult to leverage such resources due to data scarcity: There is little parallel data to these constructed languages, and creating new translation heavily depends on expert curation. Instead of relying on manually-created data, we aim to learn an artificial language in a more unsupervised fashion in parallel with training the NMT model. Specifically, our goal is to learn a sequence of tokens to represent the source sentences, which then serves as context for the NMT decoder. Table 1 illustrates this idea.

source sentence (English)	learning	a	new	language
	↓	↓	↓	↓
discrete codes	3	609	57	1042
source sentence (Indonesian)	belajar	bahasa	baru	
	↓	↓	↓	
discrete codes	3	57	258	

Table 1: We aim to learn a sequence of discrete codes to represent source sentences in multilingual NMT models. Our goal is to 1) improve inference-time robustness, 2) have more interpretable intermediate representations.

A potential advantage of representing inputs in discrete tokens is *robustness*, a property especially relevant when NMT systems must cope with unexpected testing conditions. By discretization, we restrict the continuous latent space to a finite size, providing the possibility for model intermediate representations to fall back to a position seen in training. For instance, in zero-shot translation, where the model translates directions never seen in training, the inference-time behavior is often unstable (Gu et al., 2019; Al-Shedivat and Parikh, 2019; Rios et al., 2020; Raganato et al., 2021). In practice, pivoting through an intermediate language typically gives a strong performance upper bound difficult to surpass by direct zero-shot translation (Al-Shedivat and Parikh, 2019; Arivazhagan et al., 2019a; Zhu et al., 2020; Yang et al., 2021b). Mapping the source sentences to discrete codes could act as a *pseudo*-pivoting step, which we hope to make the model more robust under zero-shot conditions.

The discrete codes also provide a new way to interpret model representations. While there are a

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 188–202
Abu Dhabi, December 7–8, 2022.

wealth of methods to analyze knowledge-sharing in multilingual NMT (Aji et al., 2020; Mueller et al., 2020; Chiang et al., 2022), they mostly either measure translation performance as a proxy, or involve sophisticated post-processing after model training, e.g. correlation scores between model hidden states (Kudugunta et al., 2019; Chiang et al., 2022), training classifiers to probe linguistic features (Liu et al., 2021a), or pruning model submodules (Kim et al., 2021). In contrast, when the model hidden states are directly associated with discrete tokens, they are directly more *interpretable*. This characteristic is especially relevant in unseen testing conditions, where it is important to pinpoint the underlying cause of model behavior.

Despite the advantages, discretizing the latent space of NMT models makes them inherently less expressive than their fully continuous counterparts. Maintaining translation performance relative to the continuous models is therefore a challenge. To strike a balance between expressiveness and discretization, we propose a *soft* discretization approach: In training, we assign each encoder hidden state to an entry in a fixed-size codebook. This step in effect clusters encoder hidden states to one of the many cluster centers in the latent space. The codebook where the cluster centers come from is then trained along with the translation model. To ensure that the decoder receives sufficient context information, we make it access both the discretized or continuous context, as illustrated in Figure 1. In our experiments on data from the Large-Scale Multilingual Translation Shared Task (Wenzek et al., 2021) from WMT21 (Akhbardeh et al., 2021), our approach is able to learn meaningful discrete codes and achieve translation performance competitive with models with continuous latent spaces. Our main contributions are:

- We propose a framework to learn discrete tokens as intermediate representations of multilingual NMT models (§3).
- On large-scale multilingual translation experiments, our approach is competitive with strong alternatives while offering more interpretable intermediate representations (§5.1).
- We use the learned discrete codes to study the role of bridging languages. Using two novel analyses, namely *code overlap* and *code translation*, we discover that using a similar bridge language facilitates knowledge-sharing in all languages covered by the model (§5.2).

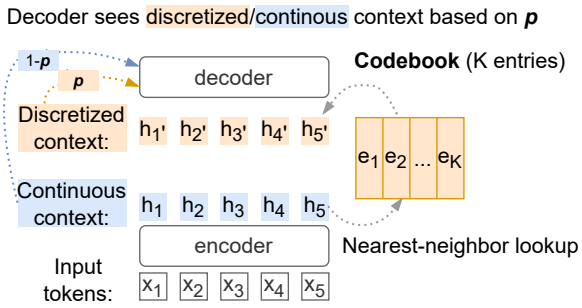


Figure 1: An illustration of our approach, which introduces a codebook for discretizing the encoder output latent space. During training, the decoder sees discretized and continuous context based on probability p. For inference, we use the continuous context, which have been well-clustered into a set of cluster centers after training.

2 Related Work

Multilingual Machine Translation Multilingual translation models are able to multitask over many language pairs. For this large-scale multi-task learning problem, training data plays a critical role. Low-resource directions often need upsampling to perform well (Arivazhagan et al., 2019b; Tang et al., 2021), which, meanwhile, brings capacity bottlenecks (Aharoni et al., 2019) to high-resource languages. This capacity bottleneck can be eliminated by dedicated language-specific capacity (Bapna and Firat, 2019; Philip et al., 2020; Shazeer et al., 2017; Zhang et al., 2021). When scaling up translation coverage (Aharoni et al., 2019; Zhang et al., 2020; Fan et al., 2021), zero-shot directions that have not seen any parallel training data is more likely to get encountered. While many dedicated models or objectives have been proposed to improve the zero-shot performance (Al-Shedivat and Parikh, 2019; Arivazhagan et al., 2019a; Pham et al., 2019; Zhu et al., 2020; Son and Lyu, 2020; Liu et al., 2021a; Yang et al., 2021b; Raganato et al., 2021), there is in general a *tradeoff* between supervised and zero-shot performance.

Robustness in Zero-Shot Conditions Zero-shot generalization is a widely-discussed direction in machine learning research (Socher et al., 2013; Norouzi et al., 2014; Romera-Paredes and Torr, 2015; Xian et al., 2017). In the context of NMT, early multilingual models already possess some capability of zero-shot translation of directions unseen in training (Ha et al., 2016; Johnson et al., 2017). However, zero-shot performance has been shown highly sensitive to, among other factors,

training data diversity (Rios et al., 2020), language token strategies (Wu et al., 2021; ElNokrashy et al., 2022), and dropout configurations (Arivazhagan et al., 2019a; Liu et al., 2021b). A main cause of the degraded quality is that the zero-shot inference generates *off-target* translation (Zhang et al., 2020) into a language other than the desired one. In recent shared tasks (Anastasopoulos et al., 2021; Libovický and Fraser, 2021a), generating synthetic data by back-translation (Sennrich et al., 2016) to eliminate zero-shot conditions has been a dominant approach for improving upon pure unsupervised settings (Pham et al., 2021; Zhang and Sennrich, 2021; Liu and Niehues, 2021; Knowles and Larkin, 2021; Libovický and Fraser, 2021b). A main motivating factor for converting zero-shot conditions to semi-supervised ones is that the latter provides more robust and consistent inference-time behavior. In this light, to fully realize the potential of knowledge-sharing in multilingual NMT, improving zero-shot robustness is an essential task.

Discrete Representations Vector Quantized Variational Autoencoder (VQ-VAE; van den Oord et al. 2017) learns discrete tokens for continuous inputs such as images and audio, and showed its effectiveness in creating discrete representations for speech representations on practical tasks (Tjandra et al., 2020; Baevski et al., 2020). Kaiser et al. (2018) proposed an improvement to VQ-VAE by *slicing*, i.e. decomposing to quantization input and output into several subspaces. The sliced variant was used in auto-encoding for learning shorter sequences, which allows to accelerate the target generation in auto-regressive decoders. The most related work to ours is probably that of Escolano et al. (2021), who used sliced VQ-VAE (Kaiser et al., 2018) on translation tasks. The main difference is that our focus is fully parameter shared multilingual systems while Escolano et al. (2021) focused on auto-encoding and bilingual systems using language-specific encoders and decoders. Therefore, in Escolano et al. (2021) zero-shot translation only occurs after a subsequent training step on dedicated encoder for the new language. Moreover, our approach extends sliced VQ-VAE (Kaiser et al., 2018) by soft codes that utilizes both continuous and quantized encoder hidden states.

3 Learning Discrete Codes

As motivated in §1, we aim to learn to represent sources sentences with a sequence of discrete codes out of a codebook. To this end, alongisde the translation objective, we also train our model to partition the continuous latent space of the encoder output into discrete subspaces. Each of the discrete subspaces is represented by one of the k entries (cluster centers) from a trainable codebook, and the encoder hidden states are assigned to these entries. To learn a meaningful discretization, the learned cluster centers must fulfill some requirements: 1) avoid trivial solutions where all points are assigned to one or a few codebook entries, 2) carry sufficient context information for the decoder for the translation task, despite being less expressive than the encoder output prior to the discretization step.

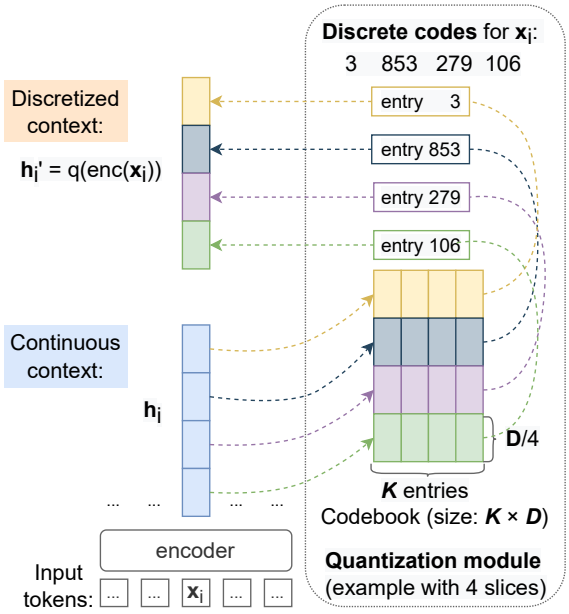


Figure 2: Illustration of the generation of the discrete codes based on a sliced (Kaiser et al., 2018) codebook.

3.1 Discretizing Encoder Latent Space

Compared to a standard Transformer (Vaswani et al., 2017), our model includes a quantization module between the encoder and decoder. We denote the quantization operation as $q(\cdot)$. Before being passed to the decoder, the encoder hidden states $\text{enc}(X)$ for input sequence X first goes through the quantization module, which runs a nearest neighbor lookup in an embedding table, i.e. the codebook. Following the notations from van den Oord et al. (2017), the codebook $e \in R^{K \times D}$ has K entries, each with dimensionality D. In our case, D is the same as the embedding dimension of the encoder, resulting in $q(\text{enc}(X))$ with the same shape as $\text{enc}(X)$.

For an input token X_i, its quantized representation is one of the K entries from the codebook $e_{k\in[1,K]}$, where k is determined by a nearest neighbor search in the embedding space, using the encoder output $\text{enc}(X_i)$ as query:

$$k = \arg\min_{j\in[k]} \|\text{enc}(X_i) - e_j\|_2, \quad (1)$$

where $\|\cdot\|_2$ indicates the Euclidean distance.

The quantization step above is vulnerable to index collapse (Kaiser et al., 2018), where only few entries from the embedding table are actively used. On auto-encoding tasks, Kaiser et al. (2018) proposed a countermeasure by breaking down the hidden dimension into multiple slices and quantizing each of them. Specifically, for input token X_i, its encoder hidden state $\text{enc}(X_i)$ is split into S slices:

$$\text{enc}(X_i)_1 \oplus \text{enc}(X_i)_2 \cdots \oplus \text{enc}(X_i)_S, \quad (2)$$

where each slice $\text{enc}(X_i)_{j\in[S]}$ is of D/S dimensions. A nearest neighbor search is conducted for each slice on the corresponding dimensions in the embedding table. The results are then concatenated and form the quantized representation:

$$q(\text{enc}(X_i)_1) \oplus q(\text{enc}(X_i)_2) \cdots \oplus q(\text{enc}(X_i)_S), \quad (3)$$

and passed to the decoder as context. Figure 2 illustrates this process.

The slicing mechanism resembles multi-head attention (Vaswani et al., 2017) in that both split the embedding dimension into subspaces for richer representation. Therefore, we will use the same number of slices as the number of attention heads.

3.2 Soft Discrete Codes

Training Compared to encoder outputs in a continuous space, the quantization module is an *information bottleneck*. In practice, limiting the amount of context information passed to the decoder will likely degrade translation quality. To strike a balance between discretization and performance, we make the discrete codes *soft*, in that the decoder can still access to the richer information prior to quantization by a probability. Specifically, during training, the encoder gives the quantized context $q(\text{enc}(X))$ by probability p, and the raw context $\text{enc}(X)$ by probability $1-p$. This procedure is illustrated in Figure 1.

In Equation 1, the lookup of index k is a non-differentiable operation. When the encoder passes on the quantized context, in order to train the parameters below the quantization module, we use the straight-through estimator (Bengio et al., 2013) to copy gradients onto the pre-quantization encoder outputs. For the copied gradients to be useful for training, the difference between $\text{enc}(X_i)$ and $q(\text{enc}(X_i))$ should be limited. To achieve this, we use the codebook loss and commitment loss from VQ-VAE (van den Oord et al., 2017):

$$\mathcal{L}_{\text{codebook}} = \|\text{sg}[\text{enc}(X)] - q(\text{enc}(X))\|_2 \quad (4)$$

and

$$\mathcal{L}_{\text{commitment}} = \|\text{enc}(X) - \text{sg}[q(\text{enc}(X))]\|_2, \quad (5)$$

where $\text{sg}[\cdot]$ denotes the stop gradient operation. Intuitively, Equation 4 pushes the codebook entries closer to the points assigned to them, while Equation 5 limits the growth of the encoder hidden states by clipping them to the codebook entries. Each of the terms has weights α_{codebook} and $\alpha_{\text{commitment}}$ to control their importance relative to the main translation objective.

Inference After training with this mechanism, one can expect that the encoder hidden states are well-clustered around a set of codebook entries. At test time, we use the continuous context $\text{enc}(X)$ which still carries more information than the cluster centers represented by the codebook entries. We will verify this property in later experiments (§6).

4 Experimental Setup

To experiment on realistic data volumes, we use the parallel data[1] from the Large-Scale Multilingual Machine Translation Shared Task (Wenzek et al., 2021) from WMT 2021 (Akhbardeh et al., 2021). We focus on small-task-2 on Southeast Asian languages. To study model robustness in zero-shot conditions and the role of language relatedness, we select parallel data between the two high-resource languages: Indonesian (id) and English (en) and three other languages in the Austronesian family: Javanese (jv), Malay (ms), and Filipino/Tagalog (tl). This leads to two data conditions:

- Indonesian-bridge (**ID-BRIDGE**)
- English-bridge (**EN-BRIDGE**)

As pretrained initialization has been shown beneficial in many submissions last year (Yang et al.,

[1] https://data.statmt.org/wmt21/multilingual-task/small_task2_filt_v2.tar.gz

	jv	ms	tl	id	en
jv		340K	662K	644K	2,556K
ms	2M		1,174K	4,060K	12,023K
tl	3M	16M		2,356K	12,348K
en	18M	230M	158M		
id	5M	65M	30M		

Table 2: Number of sentence pairs (above diagonal) and target tokens (below diagonal) from bitext for each languages pair after preprocessing. Data marked with light gray are used in the main experiments.

2021a; Liao et al., 2021; Xie et al., 2021), we initialize the models with the pretrained M2M-124 model provided in the shared task (Wenzek et al., 2021). It is worth noting that M2M-124 has seen parallel data for our *zero-shot* directions, hence zero-shot only describes the condition in our *finetuning* step. This setup is motivated by the observation that existing pretrained models are often trained on massive amounts of data, which are not always feasible to access or store. We therefore treat the pretrained M2M-124 as a given resource, without relying on all its training parallel data. We use this setup to especially study if the models can retain the pretrained knowledge on directions that are zero-shot in finetuning.

4.1 Data

The training parallel data (Wenzek et al., 2021) are compiled from the OPUS platform (Tiedemann, 2012). The specific datasets are listed in Appendix B. As parts of the training data are crawled and therefore rather noisy, we follow the filtering steps opened sourced by Fan et al. (2021), including length filtering, bitext de-duplication, and histogram filtering. An overview of the training data after filtering is in Table 2. Following the evaluation protocol of the shared task (Wenzek et al., 2021), we report spBLEU on the FLoRes-101 (Goyal et al., 2022) devtest set. We additionally report chrF++ (Popović, 2017) as another metric.

4.2 Baselines

Besides comparing to directly training on our baseline model, we also compare to two existing approaches that encourage language-independent representations, both of which have been shown effective in zero-shot translation:

Language-Independent Objective (Pham et al., 2019; Arivazhagan et al., 2019a) applies an additional loss function that enforces the representations for the source and target sentences to be similar. The loss function minimizes the difference between encoded source and target sentences after pooling. Details about the implementation are in Appendix C.1.

Adversarial Language Classifier (Arivazhagan et al., 2019a) aims to remove source language signals from the encoder hidden states, and thereby create more language-independent representations. A language classifier is trained on top of the encoder, and its classification performance is used adversarially on the encoder through a gradient reversal layer (Ganin et al., 2016). Details about the implementation are in Appendix C.2.

4.3 Training and Inference Details

As motivated in §4, we finetune from the small variant of M2M-124 with 175M parameters. This model has a vocabulary size of 256K, 6 layers in both the encoder and decoder, 16 attention heads, embedding dimension of 512 and inner dimension of 2048. As the training data for different languages are very unbalanced, we use temperature-based sampling (Arivazhagan et al., 2019b) with coefficient 5.0, which heavily upsamples low-resource directions and is recommended for unbalanced data conditions (Arivazhagan et al., 2019b; Tang et al., 2021). Additional details are in Appendix A.

For our codebook approach, we use 10K codebook entries. Initial trials with a size of 1K gave worse performance, while 40K heavily reduced training speed. We choose 16 slices[2] for the codebook, the same value as the number of attention heads. We keep these two values identical as both slicing and multi-head attention breaks the embedding dimension into multiple subspaces of lower dimensionality. The scale on the codebook loss and commitment loss (α_{codebook} and $\alpha_{\text{commitment}}$) are 1.0 and 1.001. We found the model sensitive to increasing $\alpha_{\text{commitment}}$, where higher values leads to index collapse[3]. After exponentially decreasing it to approach 1.0, we settled at 1.001. For the probability of seeing the continuous encoder context, with a search among {0.1, 0.5, 0.7 0.9}, we found 0.9 and 0.5 the best parameters for ID-BRIDGE and EN-BRIDGE respectively.

We implement our approach and the two baselines (§4.2) with FAIRSEQ (Ott et al., 2019)[4].

[2]Initial experiments on smaller datasets showed weaker translation performance with 2 and 4 slices.

[3]A potential reason is the encoder parameters are updated too aggressively by the commitment loss in these cases.

[4]Code available at: https://github.com/dannigt/

ID	Model	Avg. spBLEU(↑) (left) and chrF++(↑) (right)							
		{jv, ms, tl} →X		X→ {jv, ms, tl}		Y↔Z		Avg. (all dir.)	
	ID-BRIDGE (X=id)								
(1)	random initialization	27.5	52.7	24.2	49.4	15.8	41.5	20.8	46.3
(2)	M2M-124 (Fan et al., 2021; Goyal et al., 2022)	20.0	45.7	14.7	38.9	9.9	34.3	13.6	38.3
(3)	↪ parallel data (no data for **Y↔Z**)	27.1	52.5	24.2	49.6	17.7	43.3	21.7	47.2
(3.1)	+ language-independent objective	27.1	52.4	24.2	49.6	18.4	43.8	22.0	47.4
(3.2)	+ adversarial language classifier	27.5	52.9	24.1	49.6	18.4	44.2	22.1	47.7
(3.3)	+ codebook (ours)	27.2	52.4	23.6	49.2	18.3	44.0	21.9	47.4
	EN-BRIDGE (X=en)								
(4)	random initialization	27.0	51.1	27.8	51.6	6.8	24.5	17.1	37.9
(5)	M2M-124 (Fan et al., 2021; Goyal et al., 2022)	19.6	43.6	14.0	37.5	9.9	34.3	13.3	37.4
(6)	↪ parallel data (no data for **Y↔Z**)	28.1	51.8	27.6	51.8	5.1	20.3	16.5	36.1
(6.1)	+ language-independent objective	27.9	51.7	27.2	51.4	17.3	42.8	22.4	47.2
(6.2)	+ adversarial language classifier	27.6	51.5	27.1	51.5	17.2	42.8	22.3	47.2
(6.3)	+ codebook (ours)	26.8	50.6	26.3	50.9	15.2	39.3	20.9	45.0

Table 3: Translation quality in spBLEU(↑) and chrF++(↑). "↪" indicates finetuning on the parallel data (ID-BRIDGE or EN-BRIDGE; §4). Pivoting through the bridge language for $Y \leftrightarrow Z$ directions scores 19.7, 17.5 spBLEU and 44.9, 42.8 chrF++ for ID-BRIDGE and EN-BRIDGE respectively using the systems in rows (1) and (4).

5 Main Results

We first discuss the translation performance of our multilingual systems (§5.1), and then use the learned discrete codes to investigate cross-lingual knowledge-sharing of the trained models (§5.2).

5.1 Translation Performance

Baseline Conditions To set the upcoming results in context, we first present the performance of training without additional improvements in rows (1)-(3) and (4)-(6) of Table 3. Rows (1) and (4) show the performance of training with random initialization. This corresponds to a condition where we have parallel data but no pretrained resources. On the other side of the spectrum, in row (2) and (5), we report the results of directly running inference on the pretrained M2M-124 model. This corresponds to another extreme where we have access to pretrained models but cannot additionally train on parallel data. In rows (3) and (6), we combine the best of two worlds: initializing with pretrained model and finetuning on parallel data. For supervised directions, pretraining mainly improves →English directions: In the EN-BRIDGE condition, initializing with M2M-124 gains 1.1 spBLEU over random initialization, from 27.0 to 28.1 spBLEU. For other supervised directions, however, we do not observe gains from pretraining. This could be related to the pretrained model being particularly strong at decoding English. For zero-shot directions in our setup (these directions are seen in training by the pretrained model), as they are comparatively low-resource among all the directions covered in M2M-124, out-of-box translation quality on these directions is relatively low, with an average of 9.9 spBLEU. However, when finetuning, we see a striking difference between ID-BRIDGE and EN-BRIDGE: there is a large gain from 9.9 to 17.7 spBLEU with the former, but a degradation from 9.9 to 5.1 spBLEU for the latter. We study this phenomenon next.

Impact of Bridge Languages For EN-BRIDGE, the finetuning step causes catastrophic forgetting of the zero-shot directions (−4.8 spBLEU). On the other hand, for the ID-BRIDGE condition, pure finetuning leads to substantial improvements in *both* supervised and zero-shot directions. The gain from 9.9 to 17.7 spBLEU in the $Y \leftrightarrow Z$ directions is particularly noteworthy since the model has not seen parallel data for these directions in finetuning. This indicates that the growth in supervised directions brings zero-shot directions forward too. Moreover, on these directions, pretraining also gives large gain of 1.9 spBLEU over random initialization. Overall, the observations suggest that incorporating a similar language as bridge is beneficial to re-using pretrained knowledge. Furthermore, given that the amount of parallel data in the EN-BRIDGE condition is nearly 4 times of that in the ID-BRIDGE condition, using a similar bridge language also appears to be more *data-efficient*. This likely related to all translation directions being similar, therefore easing the multilingual learning task.

fairseq/tree/master/examples/quant

Impact of Using Codebooks Compared to pure finetuning in rows (3) and (6), by incorporating the codebook we improve zero-shot translation by 0.6 and 10.1 spBLEU for ID-BRIDGE andEN-BRIDGE respectively. Compared to the two existing approaches, namely language-independent objective and adversarial language classifier in rows (∗.1) and (∗.2), our approach performs on par with them for ID-BRIDGE, achieving 18.3 spBLEU for $Y \leftrightarrow Z$ directions and 21.9 spBLEU over all directions. In the more challenging EN-BRIDGE condition, we fall behind the two other approaches by around 2.0 spBLEU on zero-shot directions. Using a language identifier[5] (Costa-jussà et al., 2022), we found that the culprit here is still off-target translation, where some test sentences were translated to an incorrect language. While our codebook approach reduces the proportion of off-target sentences from 87.4% to 13.1% compared to the pure finetuning baseline in row (6), the figure is still higher than the 4.7% achieved by the two alternative models in rows (6.1) and (6.2). Despite this gap, an advantage of our approach is easier analyses of learned representations, which we will now leverage to investigate why the two data conditions come with very distinct zero-shot behavior.

5.2 Using Discrete Codes to Interpret Learned Representations

Since our codebook approach allows easier interpretation of model hidden representations, we take advantage of this characteristic to answer the following question: *why is the* ID-BRIDGE *data condition more performant despite using less data?*

Formalization To this end, we first extract the discrete codes for all source languages on the test set[6]. Given a total of S slices, a sentence with t tokens $X_{1,\ldots,t}$ is represented as S sets of discrete tokens $T^s_{1,\ldots,t}$ for slice s, where $s \in [S]$. Between two sets of semantically identical sentences (e.g. multiway test sets in two different languages), we can compare the discrete codes by examining: 1) their overlap and 2) the difficulty of transforming one set to another. The results quantify the similarity between the two sets of codes, and hence the model representations for the two source languages.

[5] https://github.com/facebookresearch/fairseq/tree/nllb#lid-model

[6] The FLoRes-101 test set is multiway. Therefore the semantic meanings of the sentences are the same.

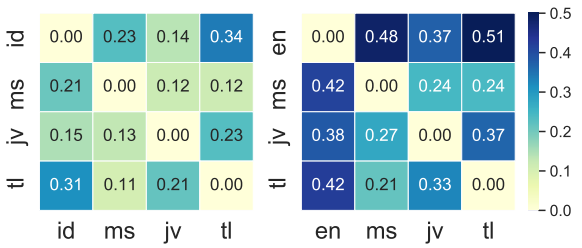


Figure 3: KL divergence(↓) of code distribution for the ID-BRIDGE (left) and EN-BRIDGE (right) setup. *Lower* values indicate a higher degree of sharing. ID-BRIDGE results in more sharing not only between itself and {ms, jv, tl} but also among {ms, jv, tl}.

Discrete Code Distribution For each slice, we normalize the code occurrences into a probability distribution. The distribution P is defined by:

$$p(c_i) = \frac{\text{frequency}(c_i)}{\sum_{c_j \in [C]} \text{frequency}(c_j)}, \quad (6)$$

where c_i is a discrete code from the set $[C]$. For a pair of languages i and j, we then compute the KL divergence between their code distributions P_i and P_j:

$$D^{(i,j)}_{\text{KL}} = (P_i || P_j). \quad (7)$$

Figure 3 depicts the KL divergence of code distribution averaged over all slices. A comparison of the En- and ID-BRIDGE setup exhibits several major differences. First, the clearly prominent first row and column in EN-BRIDGE shows that its bridge-language is represented very differently from all other languages ({ms, jv, tl}). For the ID-BRIDGE counterpart, the difference between the bridge language and the remaining languages is much milder. Second, but perhaps more importantly, among the languages used in zero-shot directions ({ms, jv, tl}), the amount of sharing is also higher under the ID-BRIDGE setup. This finding is crucial as the raw tokens for {ms, jv, tl} are identical between the ID-BRIDGE and EN-BRIDGE setup. Therefore, the higher degree of sharing is clearly an outcome of the model creating its representations differently. Overall, these results show that the choice of the bridge language not only impacts the knowledge-sharing mechanism between itself and the remaining languages, but also for the remaining languages in the model.

Discrete Code Translation The code distribution analysis above makes a simplified assumption by considering the discrete codes as a *bag of words*.

To additionally assess the *structural (dis)similarity* between the code representations for different languages, we consider the task of *translating* the discrete codes of a language to another.

While a constructed language like Interlingua would create the same representations for the source sentences with identical meanings, our discrete code representation is not yet invariant to the source language. Nevertheless, we do expect them to be more abstracted from the source sentences, making the translation task easier than directly between the raw tokens. Here we train a translation model on the discrete codes and use the test performance to quantify how similarly the source languages are represented. When the representations are more different from each other, i.e. language-specific, the translation quality on the discrete is expected to be lower.

Specifically, we randomly sample 100K sentence pairs[7] for each translation direction in the experiments of Table 3 extract their discrete codes assigned by the trained models (rows (3.3) and (6.3) of Table 3), and train a new Transformer-base (Vaswani et al., 2017) to translate between the extracted codes of different languages. We flatten the slices, therefore making each source token represented by 16 discrete codes. After training for 200K steps, we report BLEU scores on the test set, which is also converted to discrete codes. The results are shown in Figure 4. First, the translation task is clearly easier on the discrete codes derived from the ID-BRIDGE system. Second, the scores differences are especially prominent when translating out of Malay (ms) and Javanese (jv), which are more related to Indonesian than Filipino/Tagalog (tl). Along with the results from the code overlap, our results show that using a similar bridge language results in higher knowledge-sharing not only syntactically but also structurally, especially between related languages.

6 Analyses on Learned Discrete Codes

Next we further investigate the discrete codes regarding its usefulness for the learned representations (§6.1) as well as the translation task (§6.2).

6.1 How well-clustered are the hidden states?

As motivated in §3, although at inference time we use the continuous encoder hidden states instead of

[7]The training data (Table 2) allow us to use 340K sentences. We sampled 100K for faster experiment iteration.

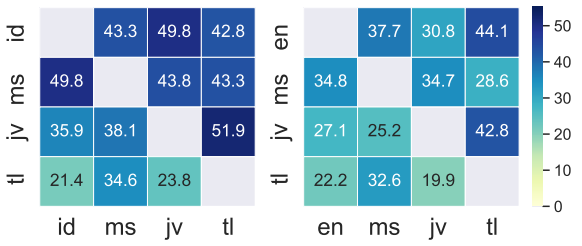
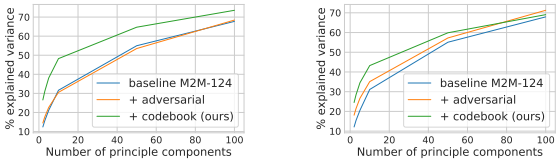


Figure 4: BLEU(↑) scores of translating between discrete codes for the ID-BRIDGE (left) and EN-BRIDGE (right) setup. *Higher* values indicate a higher degree of sharing. In general it is easier to translate the codes for the ID-BRIDGE setup, indicating more structural similarity between the representations.



(a) ID-BRIDGE (b) EN-BRIDGE

Figure 5: Our codebook approach creates better-clustered encoder hidden states, as shown by a much higher percentage of variance explained by PCA compared to both the baseline and a strong alternative approach (adversarial language classifier).

the cluster centers, the soft discrete codes will still enforce encoder hidden states into clusters, thereby resembling a discrete structure. To verify whether the encoder latent space indeed becomes more discretized with our approach, we analyze the encoder hidden states on the test set using Principle Component Analysis (PCA). If the data points representing the encoder outputs are well-clustered, a larger percentage of their variance should be explained by the learned principle components. As shown in Figure 5, our approach (marked with green line) consistently leads to higher proportions of explained variances compared to the baseline `M2M-124`, as well as the strong alternative approach with the adversarial language classifier. These results therefore confirm the effectiveness of our soft discrete code approach in enforcing discrete structures in the encoder latent space.

6.2 Meaningfulness of Clusters Centers

Recall that at inference time our soft discrete code model uses the encoder hidden states prior to discretiztaion, although it does use both pre- and post-discretization encoder context in training. A main reason of doing so is that discretizing the encoder

hidden states to cluster centers creates an information bottleneck that limits model expressiveness. Despite the expected performance degradation, we are nonelessness interested in *quantifying* how much information is lost by using the cluster centers as context instead. In other words, the question is *how meaningful are the cluster centers for the translation task?* In Table 4, we report the results of using the cluster centers as context for the decoder at inference time. Compared to using the encoder hidden states, we see a degradation of 4.1 and 1.7 and spBLEU for ID-BRIDGE and EN-BRIDGE respectively. This indicates that the cluster centers are still relevant for the translation task, although much less powerful than the encoder hidden states prior to discretization. It also rules out the possibility of the learned codes being trivial repetitions, which would otherwise have been detrimental to the translation performance.

Encoder States at Inference	Avg. spBLEU(↑)			
	→X	X→	Y↔Z	Avg.
ID-BRIDGE (X=id)				
encoder states (Tab. 3 row (3.3))	27.2	23.6	18.3	21.9
cluster centers	22.8	20.0	14.3	17.8
EN-BRIDGE (X=en)				
encoder states (Tab. 3 row (6.3))	26.8	26.3	15.2	20.9
cluster centers	24.3	24.6	13.9	19.2

Table 4: At inference time, using cluster centers instead of the clustered encoder states degrades performance by 1.7-4.1 spBLEU. Despite the degradation, the scores show that translation from the clusters centers is still meaningful. This also rules out the possibility of the learned codes collapsing to trivial repetitions.

7 Analyses on Zero-Shot Translation

Our experiments so far use single-bridge languages and are evaluated in part on zero-shot directions. We now study the impact when either of the two conditions changes: 1) when parallel data is available for previously zero-shot directions; 2) when using multiple bridge languages.

7.1 When does zero-shot translation match the performance on parallel data?

Zero-shot conditions could be avoided by creating synthetic data from back-translation (Sennrich et al., 2016; Zhang et al., 2020) or mining additional parallel data (Fan et al., 2021; Freitag and Firat, 2020). Both approaches introduce additional workflows into the pipeline of building translation systems. We are therefore interested in the following question: *How much parallel data do we need to perform better than direct zero-shot translation?*

The training corpora from the shared task (§4.1) provides an oracle condition to answer this question. As shown in Table 2, the oracle parallel data amounts to 2.2M sentences in total (340K for jv-ms, 662K for jv-tl, and 1.2M sentences for ms-tl). We take 100%, 10% and 1% of the oracle parallel data and training systems together with the original data and train multilingual systems with the same configuration as rows (3) and (6) of Table 3. The results are shown in Table 5.

To our surprise, adding 1% oracle bitext (22K sentence pairs in total) of the previously zero-shot directions already results in comparable performance to the best zero-shot performance (18.4 and 17.3 spBLEU for ID-BRIDGE and EN-BRIDGE respectively). However, this comes with some degradation on supervised directions of 0.4 spBLEU for ID-BRIDGE and 0.7 for EN-BRIDGE. This is likely due to the temperature-based sampling aggressively upsampling the extremely low-resource directions, meanwhile causing the model to deprioritize other higher-resource directions. When increasing oracle bitext to 10% (220K sentence pairs in total), the system outperforms direct zero-shot performance. Lastly, the additional gain appear to diminish when going from 10% to all oracle data. For ID-BRIDGE, the performance appears saturated at 10%: adding the remaining 90% parallel data does not give additional gain. On the contrary, For EN-BRIDGE, the system appears to still improve, especially on $Y \leftrightarrow Z$ directions (+0.5 spBLEU). The performance on these directions nevertheless still falls behind the ID-BRIDGE direction by 0.8 spBLEU (18.9 vs 19.7 spBLEU). An explanation is that the EN-BRIDGE system requires more data to train as a result of the bridge language being very distant to the rest, thereby increasing the difficulty of multitasking over all the translation directions. This echos with the previous finding that using related bridge languages eases the multilingual translation task and increases knowledge-sharing (§5.1).

7.2 Do multiple bridge languages bring additional gains?

While the experiments so far are based on single bridge languages, in practice we often have access to multi-bridge parallel data. Indeed, recent

Oracle Bitext	Avg. spBLEU(↑)			
	→X	X→	Y↔Z	Avg.
ID-BRIDGE (X=id)				
best zero-shot (Tab. 3 row (3.2))	27.5	24.1	18.4	22.1
1%	26.9	23.9	18.4	21.9
10%	27.3	24.5	19.7	22.8
100% (2.2M bitext)	26.6	24.8	19.7	22.7
EN-BRIDGE (X=en)				
best zero-shot (Tab. 3 row (6.1))	27.9	27.2	17.3	22.4
1%	27.0	26.8	17.1	22.0
10%	27.5	27.4	18.4	22.9
100% (2.2M bitext)	27.7	27.5	18.9	23.2

Table 5: Impact of adding oracle parallel data for the previously zero-shot directions. Adding 10% parallel data (roughly 220K sentence pairs in our case) surpasses the best performance on direct zero-shot translation.

Data Condition		Avg. spBLEU(↑)			
		→X	X→	Y↔Z	Avg.
MULTI-BRIDGE	X= id	27.0	24.3	18.3	21.9
	X= en	27.8	27.7		23.0
Only ID-BRIDGE (Tab. 3 row (3))		27.1	24.2	17.7	21.7
Only EN-BRIDGE (Tab. 3 row (6))		28.1	27.6	5.1	16.5

Table 6: Results of using multiple bridges (combining ID-BRIDGE and EN-BRIDGE). Despite substantial gains over EN-BRIDGE, the multi-bridge system only gives a mild improvement in zero-shot performance ($Y \leftrightarrow Z$) over the ID-BRIDGE system.

works (Freitag and Firat, 2020; Fan et al., 2021) have shown success on large-scale fully-connected models, as well as evidence of multi-bridge outperforming the English-bridge condition (Rios et al., 2020). What remains unclear is whether there is a synergy when combining the parallel data from several single-bridge conditions. We investigate this hypothesis by training a multi-bridge system, combing the data from our ID-BRIDGE and EN-BRIDGE setup. As shown in Table 6, for supervised directions of $\rightarrow X$ and $X \rightarrow$, there is no clear difference between the performance of the multi-bridge system and that of the single-bridge ones. For zero-shot directions ($Y \leftrightarrow Z$), while multi-bridge gains substantially over EN-BRIDGE (18.3 from 5.1 spBLEU), there is only a slight gain over ID-BRIDGE. Given that the multi-bridge model more than doubles the training time of ID-BRIDGE, the little performance difference to the multi-bridge system shows that choosing a bridge language related to the remaining languages is a data-efficient way to achieve strong zero-shot performance.

8 Conclusion

In this work, we focus on learning to represent source sentences of multilingual NMT models by discrete codes. On multiple large-scale experiments, we show that our approach not only increase the model robustness in zero-shot conditions, but also offers more interpretable intermediate representations. We leverage the latter property to investigate the role of bridge languages, and show that using a more related bridge language leads to increased knowledge-sharing, not only between the bridge language and remaining but also between all other languages involved in training.

A limitation is that the discrete codes only give a mechanism to compare hidden representations, but are not directly interpretable by humans. A potential improvement would be to use an existing codebook that corresponds to an actual human language. Besides this, as next steps, we plan to improve the generation process of the discrete codes. The first direction is to make the code lookup conditionally-dependent along the time dimension and learn to shrink the sequence length of the discrete codes, thereby creating a more compact representation. Another direction is to explicitly incentivize more shared codes between different, and especially related, languages during training. This would bring the discrete codes closer to a language-independent representation.

Acknowledgement We thank James Cross and Paco Guzmán for their feedback in the early stage of this work. We thank the anonymous reviewers for their detailed and insightful feedback. We also thank Ngoc Quan Pham, Tu Anh Dinh, and Sai Koneru for feedback on the paper draft.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The AMARA corpus: Building parallel language resources for the educational domain. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1856–1862, Reykjavik, Iceland. European Language Resources Association (ELRA).

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,

Minneapolis, Minnesota. Association for Computational Linguistics.

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Maruan Al-Shedivat and Ankur Parikh. 2019. Consistency by agreement in zero-shot 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 1184–1197, Minneapolis, Minnesota. Association for Computational Linguistics.

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alexander Waibel, Changhan Wang, and Matthew Wiesner. 2021. FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 1–29, Bangkok, Thailand (online). Association for Computational Linguistics.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federmann, Dmitriy Genzel, Franscisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. TICO-19: the translation initiative for COvid-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*, Online. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019a. The missing ingredient in zero-shot neural machine translation. *CoRR*, abs/1903.07091.

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

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432.

Ting-Rui Chiang, Yi-Pei Chen, Yi-Ting Yeh, and Graham Neubig. 2022. Breaking down multilingual machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2766–2780, Dublin, Ireland. Association for Computational Linguistics.

Christos Christodoulopoulos and Mark Steedman. 2015. A massively parallel corpus: the bible in 100 languages. *Lang. Resour. Evaluation*, 49(2):375–395.

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, Loïc 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. *CoRR*, abs/2207.04672.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational*

Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Muhammad N. ElNokrashy, Amr Hendy, Mohamed Maher, Mohamed Afify, and Hany Hassan Awadalla. 2022. Language tokens: A frustratingly simple approach improves zero-shot performance of multilingual translation. *CoRR*, abs/2208.05852.

Carlos Escolano, Marta R. Costa-Jussà, and José A. R. Fonollosa. 2021. From bilingual to multilingual neural-based machine translation by incremental training. *Journal of the Association for Information Science and Technology*, 72(2):190–203.

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, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *The Journal of Machine Learning Research*, 22:107:1–107:48.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Markus Freitag and Orhan Firat. 2020. Complete multilingual neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 550–560, Online. Association for Computational Linguistics.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1258–1268, Florence, Italy. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alex Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. In *Proceedings of the 13th International Conference on Spoken Language Translation*, Seattle, Washington D.C. International Workshop on Spoken Language Translation.

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.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2395–2404. PMLR.

Zae Myung Kim, Laurent Besacier, Vassilina Nikoulina, and Didier Schwab. 2021. Do multilingual neural machine translation models contain language pair specific attention heads? In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2832–2841, Online. Association for Computational Linguistics.

Rebecca Knowles and Samuel Larkin. 2021. NRC-CNRC systems for Upper Sorbian-German and Lower Sorbian-German machine translation 2021. In *Proceedings of the Sixth Conference on Machine Translation*, pages 999–1008, Online. Association for Computational Linguistics.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Baohao Liao, Shahram Khadivi, and Sanjika Hewavitharana. 2021. Back-translation for large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 418–424, Online. Association for Computational Linguistics.

Jindřich Libovický and Alexander Fraser. 2021a. Findings of the WMT 2021 shared tasks in unsupervised

MT and very low resource supervised MT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.

Jindřich Libovický and Alexander Fraser. 2021b. The LMU Munich systems for the WMT21 unsupervised and very low-resource translation task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 989–994, Online. Association for Computational Linguistics.

Danni Liu and Jan Niehues. 2021. Maastricht university's multilingual speech translation system for IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 138–143, Bangkok, Thailand (online). Association for Computational Linguistics.

Danni Liu, Jan Niehues, James Cross, Francisco Guzmán, and Xian Li. 2021a. Improving zero-shot translation by disentangling positional information. 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 1259–1273, Online. Association for Computational Linguistics.

Hui Liu, Danqing Zhang, Bing Yin, and Xiaodan Zhu. 2021b. Improving pretrained models for zero-shot multi-label text classification through reinforced label hierarchy reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1051–1062, Online. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Arya D. McCarthy, Dylan Lewis, Winston Wu, and David Yarowsky. 2020. An analysis of massively multilingual neural machine translation for low-resource languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3710–3718, Marseille, France. European Language Resources Association.

Mohammad Norouzi, Tomás Mikolov, Samy Bengio, Yoram Singer, Jonathon Shlens, Andrea Frome, Greg Corrado, and Jeffrey Dean. 2014. Zero-shot learning by convex combination of semantic embeddings. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

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.

Ngoc-Quan Pham, Tuan Nam Nguyen, Thanh-Le Ha, Sebastian Stüker, Alexander Waibel, and Dan He. 2021. Multilingual speech translation KIT @ IWSLT2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 154–159, Bangkok, Thailand (online). Association for Computational Linguistics.

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha, and Alexander Waibel. 2019. Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 13–23, Florence, Italy. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online. 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.

Alessandro Raganato, Raúl Vázquez, Mathias Creutz, and Jörg Tiedemann. 2021. An empirical investigation of word alignment supervision for zero-shot multilingual neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8449–8456, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Annette Rios, Mathias Müller, and Rico Sennrich. 2020. Subword segmentation and a single bridge language affect zero-shot neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 528–537, Online. Association for Computational Linguistics.

Bernardino Romera-Paredes and Philip H. S. Torr. 2015. An embarrassingly simple approach to zero-shot learning. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2152–2161. JMLR.org.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2021. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1351–1361, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 935–943.

Bokyung Son and Sungwon Lyu. 2020. Sparse and decorrelated representations for stable zero-shot NMT. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2260–2266, Online. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. 2020. Transformer VQ-VAE for unsupervised unit discovery and speech synthesis: Zerospeech 2020 challenge. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 4851–4855. ISCA.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6306–6315.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Guillaume Wenzek, Vishrav Chaudhary, Angela Fan, Sahir Gomez, Naman Goyal, Somya Jain, Douwe Kiela, Tristan Thrush, and Francisco Guzmán. 2021. Findings of the WMT 2021 shared task on large-scale multilingual machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 89–99, Online. Association for Computational Linguistics.

Liwei Wu, Shanbo Cheng, Mingxuan Wang, and Lei Li. 2021. Language tags matter for zero-shot neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3001–3007, Online. Association for Computational Linguistics.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning - the good, the bad and the ugly. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3077–3086. IEEE Computer Society.

Wanying Xie, Bojie Hu, Han Yang, Dong Yu, and Qi Ju. 2021. TenTrans large-scale multilingual machine translation system for WMT21. In *Proceedings of the Sixth Conference on Machine Translation*, pages 439–445, Online. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021a. Multilingual machine translation systems from Microsoft for WMT21 shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 446–455, Online. Association for Computational Linguistics.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. 2021b. Improving multilingual translation by representation and gradient regularization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7266–7279, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Biao Zhang and Rico Sennrich. 2021. Edinburgh's end-to-end multilingual speech translation system for IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 160–168, Bangkok, Thailand (online). Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1628–1639, Online. Association for Computational Linguistics.

Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655, Online. Association for Computational Linguistics.

A Additional Training and Inference Details

When training, one optimization step happens after 16384 tokens. We use the Adam optimizer with betas $(0.9, 0.98)$. The learning rate is 0.0001 with the inverse squared root schedule and 2500 warmup steps. As for regularization parameters, we use label smoothing of 0.1, dropout of 0.3, and attention dropout 0.1. The models are trained for 500K updates in total. An exception is the MULTI-BRIDGE experiment with more training data, where we trained for 800K updates in total. For inference, we decode with a beam size of 5.

B Dataset Details

The training parallel data include the following corpora: bible-uedin (Christodoulopoulos and Steedman, 2015), (Multi)CCAligned (El-Kishky et al., 2020), Gnome[8], ELRC[9], KDE4[10], GlobalVoices[11], OpenSubtitles[12], QED (Abdelali et al., 2014), MultiParaCrawl[13], TED2020[14], Tanzil[15], Tatoeba[16], Ubuntu[17], WikiMatrix (Schwenk et al., 2021), wikimedia[18], and TICO-19 (Anastasopoulos et al., 2020).

C Implementation of Baselines

C.1 Language-Independent Objective

We chose meanpool and L2 distance for the similarity loss since it gave better or more consistent performance in initial experiments. As for the weight of the language-independent objective, we used 1.0 following Pham et al. (2019).

[8] https://opus.nlpl.eu/GNOME.php
[9] https://opus.nlpl.eu/ELRC.php
[10] https://opus.nlpl.eu/KDE4.php
[11] https://opus.nlpl.eu/GlobalVoices.php
[12] https://opus.nlpl.eu/OpenSubtitles-v2018.php
[13] https://opus.nlpl.eu/MultiParaCrawl.php
[14] https://opus.nlpl.eu/TED2020.php
[15] https://opus.nlpl.eu/Tanzil.php
[16] https://opus.nlpl.eu/Tatoeba.php
[17] https://opus.nlpl.eu/Ubuntu.php
[18] https://opus.nlpl.eu/wikimedia.php

C.2 Adversarial Classifier

We extend the adversarial language classification approach from Arivazhagan et al. (2019a) for robust training. Specifically, we use a modified loss when adversarially training the encoder. Moreover, we apply the language classification on the token level to remove the need for selecting a pooling method. The classifier minimizes the cross-entropy loss when predicting the language labels:

$$\mathcal{L}_{\text{classifier}} = -\sum_{c=1}^{L} y_c \log(p_c), \tag{8}$$

where L is the number of classes to predict, y_c is a binary indicator whether the true language label is c, and p_c is the predicted probability for the instance belonging to language c.

Removing source language signals from the encoder representations can be achieved by a gradient reversal layer (Ganin et al., 2016) from the language classification. An issue with the standard classification loss in Equation 8 is that, when the classifier is performing well, the loss landscape is rather flat, causing minimal gradient flow to the encoder. In fact, when the classifier predicts the source languages accurately, we instead need large gradients to update the encoder representations as they contain high amounts of language signals. Therefore, when updating the encoder parameters adversarially, we use the modified loss:

$$\mathcal{L}_{\text{adv\_classifier}} = \sum_{c=1}^{L} y_c \log(1 - p_c), \tag{9}$$

which in effect mirrors Equation 8 by the horizontal axis and the vertical line defined by $x = 0.5$. With the modified loss, the optimization direction does not change, but the gradient is larger when the classifier is performing well.

The translation model is then trained with:

$$\mathcal{L}_{\text{encoder\_decoder}} = \mathcal{L}_{\text{MT}} + \mathcal{L}_{\text{adv\_classifier}}. \tag{10}$$

For training stability, we alternate the optimization of the classifier (Equation 8) and the main encoder-decoder parameters (Equation 10). Optimizing them jointly would otherwise lead to co-adaptation of the parameters of the translation and classification module and empirically causes training instability.