On the Off-Target Problem of Zero-Shot Multilingual Neural Machine Translation

Liang Chen[1] **Shuming Ma**[2] **Dongdong Zhang**[2] **Furu Wei**[2] **Baobao Chang**[1†]
National Key Laboratory for Multimedia Information Processing, Peking University[1]
Microsoft Research[2]
leo.liang.chen@outlook.com chbb@pku.edu.cn
{shumma,dozhang,fuwei}@microsoft.com

Abstract

While multilingual neural machine translation has achieved great success, it suffers from the off-target issue, where the translation is in the wrong language. This problem is more pronounced on zero-shot translation tasks. In this work, we find that failing in encoding discriminative target language signal will lead to off-target and a closer lexical distance (i.e., KL-divergence) between two languages' vocabularies is related with a higher off-target rate. We also find that solely isolating the vocab of different languages in the decoder can alleviate the problem. Motivated by the findings, we propose Language Aware Vocabulary Sharing (LAVS), a simple and effective algorithm to construct the multilingual vocabulary, that greatly alleviates the off-target problem of the translation model by increasing the KL-divergence between languages. We conduct experiments on a multilingual machine translation benchmark in 11 languages. Experiments show that the off-target rate for 90 translation tasks is reduced from 29% to 8%, while the overall BLEU score is improved by an average of 1.9 points without extra training cost or sacrificing the supervised directions' performance. We release the code at https://github.com/PKUnlp-icler/Off-Target-MNMT for reproduction.

1 Introduction

Multilingual NMT makes it possible to do the translation among multiple languages using only one model, even for zero-shot directions (Johnson et al., 2017; Aharoni et al., 2019). It has been gaining increasing attention since it can greatly reduce the MT system's deployment cost and enable knowledge transfer among different translation tasks, which is especially beneficial for low-resource languages. Despite its success, off-target is a harsh and widespread problem during zero-shot translation in existing multilingual models.

Target \ Source	cs	fr	de	fi	lv	et	ro	hi	tr	gu	AVG
cs		43%	45%	21%	13%	11%	13%	12%	10%	33%	22%
fr	20%		30%	22%	18%	21%	12%	10%	15%	12%	18%
de	15%	38%		19%	13%	16%	14%	36%	28%	36%	23%
fi	14%	32%	28%		12%	9%	13%	44%	19%	64%	26%
lv	8%	34%	24%	7%		5%	10%	33%	19%	58%	22%
et	16%	32%	15%	8%	15%		23%	47%	23%	74%	28%
ro	2%	2%	3%	2%	0%	3%		10%	8%	50%	9%
hi	15%	13%	6%	13%	20%	14%	16%		54%	78%	25%
tr	2%	1%	0%	1%	0%	1%	18%	33%		70%	14%
gu	77%	60%	53%	84%	80%	74%	80%	92%	95%		77%
AVG	19%	28%	23%	19%	19%	17%	22%	35%	30%	53%	**29%**

Table 1: Zero-shot off-target rate of the model with traditional vocab sharing on WMT'10 dataset. High values are in red and low values are in blue. The average OTR of 90 zero-shot directions is about 29%.

For the zero-shot translation directions, the model translates the source sentence to a wrong language, which severely degrades the system's credibility. As shown in Table 1, the average off-target rate on 90 directions is 29% and even up to 95% for some language pair (tr->gu) on WMT'10 dataset.

Researchers have been noticing and working on solving the problem from different perspectives. For model trained on English-centric dataset, a straight forward method is to add pseudo training data on the zero-shot directions through back-translation (Gu et al., 2019; Zhang et al., 2020). Adding pseudo data is effective since it directly turns zero-shot translation into a weakly supervised task. Despite its effectiveness, it brings a lot more training cost during generating data and training on the augmented corpus and the supervised directions' performance is also reported to decrease due to the model capacity bottleneck (Zhang et al., 2020; Yang et al., 2021). Rios et al. (2020) finds that instead of regarding all languages as one during the vocabulary building process, language-specific BPE can alleviate the off-target problem, yet it still costs the supervised directions' performance.

In this work, we perform a comprehensive analysis of the off-target problem, finding that failure in encoding discriminative target language signal

†Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 9542–9558

will lead to off-target and we also find a strong correlation between off-target rate of certain direction and the lexical similarity between the involved languages. A simple solution by separating the vocabulary of different languages in the decoder can decrease lexical similarity among languages and it proves to improve the zero-shot translation performance. However, it also greatly increases the model size (308M->515M) because a much larger embedding matrix is applied to the decoder.

For a better performance-cost trade-off, we further propose Language-Aware Vocabulary Sharing (LAVS), a novel algorithm to construct the multilingual vocabulary that increases the KL-divergence of token distributions among languages by splitting particular tokens into language-specific ones.

LAVS is simple and effective. It does not introduce any extra training cost and maintains the supervised performance. Our empirical experiments prove that LAVS reduces the off-target rate from 29% to 8% and improves the BLEU score by 1.9 points on the average of 90 translation directions. Together with back-translation, the performance can be further improved. LAVS is also effective on larger dataset with more languages such as OPUS-100 (Zhang et al., 2020) and we also observe that it can greatly improve the English-to-Many performance (+0.9 BLEU) in the large-scale setting.

2 Related Work

Off-Target Problem in Zero-Shot Translation Without parallel training data for zero-shot directions, the MNMT model is easily caught up in off-target problem (Ha et al., 2016; Aharoni et al., 2019; Gu et al., 2019; Zhang et al., 2020; Rios et al., 2020; Wu et al., 2021; Yang et al., 2021) where it ignores the target language signal and translates to a wrong language. Several methods are proposed to eliminate the off-target problem. Zhang et al. (2020); Gu et al. (2019) resort different back-translation techniques to generate data for non-English directions. Back-translation method is straight-forward and effective since it provides pseudo data on the zero-shot directions but it brings a lot more additional cost during generating data and training on the augmented corpus. Gu et al. (2019) introduced decoder pretraining to prevent the model from capturing spurious correlations, Wu et al. (2021) explored how language tag settings influence zero-shot translation. However, the cause for off-target still remains underexplored.

Vocabulary of Multilingual NMT Vocabulary building method is essential for Multilingual NMT since it decides how texts from different languages are turned into tokens before feeding to the model. Several word-split methods like Byte-Pair Encoding (Sennrich et al., 2016), Wordpiece (Wu et al., 2016) and Sentencepiece (Kudo and Richardson, 2018), are proposed to handle rare words using a limited vocab size. In the background of multilingual NMT, most current studies and models (Conneau et al., 2019; Ma et al., 2021; team et al., 2022) regard all languages as one and learn a shared vocabulary for different languages. Xu et al. (2021a) adopted optimal transport to find the vocabulary with most marginal utility. Chen et al. (2022) study the relation between vocabulary sharing and label smoothing for NMT. Closely related to our work, Rios et al. (2020) finds that training with language-specific BPE that allows token overlap can improve the zero-shot scores at the cost of supervised directions' performance and a much larger vocab while our method does not bring any extra cost.

To the best of our knowledge, we are the first to explore how vocabulary similarity of different languages affects off-target in zero-shot MNMT and reveal that solely isolating vocabulary in the decoder can alleviate the off-target problem without involving extra training cost or sacrificing the supervised directions' performance.

3 Delving into the Off-Target Problem

3.1 Multilingual NMT System Description

We adopt the Transformer-Big (Vaswani et al., 2017) model as the baseline model. For multilingual translation, we add a target language identifier <XX> at the beginning of input tokens to combine direction information. We train the model on an English-centric dataset WMT'10 (Callison-Burch et al., 2010). Zero-shot translation performance is evaluated on Flores-101 (Goyal et al., 2021) dataset. We use a public language detector[1] to identify the sentence-level language and compute the off-target rate (OTR) which denotes the ratio of translation that deviates to wrong languages. Full information about training can be found in Section 5.1.

3.2 Off-Target Statistics Safari

Off-Target Rate Differs in Directions We first train the multilingual NMT model in 10 EN-X directions and 10 inverse directions from WMT'10

[1]https://github.com/Mimino666/langdetect

An Off-Target Case
Direction: FR -> DE
Input: <DE> Un sondage effectué auprès de 1 400 personnes avant les élections fédérales de 2010 a révélé que le nombre d'opposants à la transformation de l'Australie en république avait augmenté de 8 % depuis 2008.
Output: A survey of 1400 people prior to the 2010 federal elections revealed that the number of opponents of Australia's transformation into a republic had increased by 8 % since 2008.
Gold: Von den 1.400 Personen, die vor den Bundeswahlen 2010 befragt wurden, hat der Anteil derjenigen, die sich dagegen aussprechen, dass Australien zur Republik wird, seit 2008 um 8 Prozent zugenommen.

Figure 1: A real Off-Target case observed in our multilingual NMT system. In this case, the output is literally English while the real target is German.

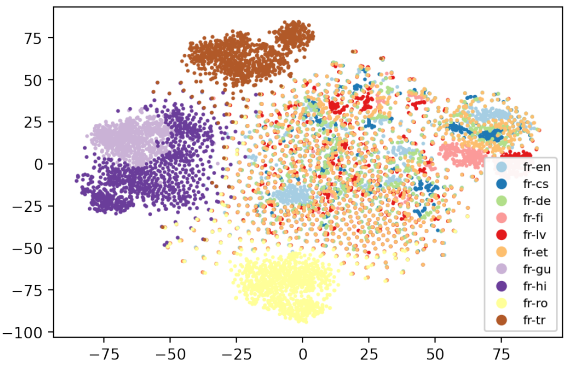


Figure 2: Encoder pooled output visualization using TSNE for French-to-Many translations. The input French sentences are the same for all directions. Note that there are only French sentences in the encoder side.

simultaneously. Then we test the model on 90 X-Y zero-shot directions using semantic parallel sentences from the previous 10 languages provided by Flores-101. We compute the off-target rate of all directions and list the result in Table 1.

In addition to the individual score, we next split the languages into High (cs, fr, de, fi), Mid (lv, et), and Low (ro, tr, hi, gu) resources according to data abundance degree. Then we compute the average OTR of High-to-High, High-to-Low, Low-to-High, and Low-to-Low directions and rank the result. The ranked result is: Low-to-Low (50.28%) > High-to-High (27.16%) > Low-to-High (23.18%) > High-to-Low (20.78%). Based on the observation, we can see that language with the lowest resource (gu) contributes to a large portion of off-target cases. This is reasonable since the model might not be familiar with the language identifier <GU> and the same situation goes for Low-to-Low translations.

However, it is surprising to see that translations between high-resource languages suffer from more severe off-target than those directions involving one low-resource language. There seem to be other factors influencing the off-target phenomena.

In other words, if data imbalance is not the key factor for off-targets between high-resource languages, what are the real reasons and possible solutions? To answer these questions, we need to delve deeper into the real off-target cases.

The Major Symptom of Off-Target When the model encounters an off-target issue, a natural question is which language the model most possibly deviates to. We find that among different directions, a majority (77%) of the off-target cases are wrongly translated to English, which is the centric language in the dataset. A small part (15%) of cases copy the the input sentence as output. Our observation also agrees with the findings of Zhang et al. (2020). It raises our interest that why most off-target cases deviate to English.

3.3 Failing in Encoding Discriminative Target Language Signal Leads to Off-Target

Considering the encoder-decoder structure of the model, we hypothesize that:

The encoder fails to encode discriminative target language information to the hidden representations before passing to the decoder.

To test the hypothesis, we start by analyzing the output of the trained transformer's encoder:

1) We choose French as the source language and conduct a French-to-Many translation (including all languages in WMT'10) on Flores-101.

2) We collect all the pooled encoder output representations of the French-to-Many translation and project them to 2D space using TSNE. The visualization result is shown in Figure 2.

The visualization result justifies our hypothesis. We can tell from the distribution that only representations belonging to "fr-tr" and "fr-ro" directions have tight cluster structures with boundaries. *The representations from high/mid-resource language pairs are completely in chaos and they are also mixed with fr-en representations.* And those languages generally have a higher off-target rate in French-to-Many Translation according to Table 1.

The decoder cannot distinguish the target language signal from the encoder's output when it receives representations from the "chaos" area. Moreover, during the training process, the decoder generates English far more frequently than other lan-

guages and it allocates a higher prior for English.

Passing hidden representation similar to English one will possibly confuse the decoder to generate English no matter what the given target language is. It could explain why most off-target cases deviate to English. The decoder struggles to tell the correct direction from the encoder's output.

Now we have a key clue for the off-target issue. The left question is what causes the degradation of target language signal in some directions and whether we can make the representations of different target languages more discriminative to eliminate the off-target cases.

3.4 Language Proximity Correlates with Zero-Shot Off-Target Rate

To explore how off-target occurs differently in different language pairs, we conduct experiments using a balanced subset of WMT'10 dataset where we hope to preclude the influence of data size. We randomly sampled 500k sentences from different directions to form a balanced training set and remove the directions(hi, tr and gu) that do not have enough sentences.

Language Proximity is an Important Characteristic of Translation Direction Our motivation is intuitive that if two languages are rather close, the probability distribution of different n-grams in the two languages' tokenized corpus should be nearly identical. Considering a large number of different n-grams in the corpus, we only consider 1-grams to compute the distribution. We call the result "Token Distribution".

We use Kullback–Leibler divergence from Token Distribution of Language B to Language A to reflect the degree of difficulty if we hope to encode sentence from B using A, which can also be interpreted as "Lexical Similarity".

$$D_{\mathrm{KL}}(A\|B) = \sum_{x\in\mathcal{V}} A(x)\log\left(\frac{A(x)}{B(x)}\right) \quad (1)$$

where $\mathcal{V}$ denotes the shared vocabulary, $A(x)$ is the probability of token x in language A. To avoid zero probability during computing Token Distribution, we add 1 to the frequency of all tokens in the vocabulary as a smoothing factor.

Lexical Similarity is related to Off-Target Rate We compute the KL divergence between language pairs with the training data. After training on the balanced dataset, the zero-shot translation is conducted on the Flores-101 dataset. We visualize the result of the top-3 languages(fr,cs,de) with most resources in WMT'10 dataset for analysis.

As shown in Figure 3, we can observe from the statistics that language proximity is highly related to the off-target rate. The Pearson correlation coefficients between the off-target rate and the KL-Divergence from target to source of the three x-to-many translations are -0.75±0.02, -0.9±0.03 and -0.92±0.03. The average Pearson correlation of all x-to-many directions is -0.77±0.11. It indicates that language pair which has higher lexical similarity from target to source may have a higher chance to encounter off-target than those language pairs which has less similar languages.

3.5 Shared Tokens in the Decoder Might Bias the Zero-Shot Translation Direction

Previous section shows a correlation between the lexical similarity and off-target rate within certain language pair. We are more interested in whether the lexical similarity will cause the representation degradation in Figure 2, which further causes off-target. In fact, larger lexical similarity suggests more shared tokens between languages and will let the decoder output more overlapped tokens during supervised training. **The token overlap for different target in output space is harmful for zero-shot translation.** During training, the decoder might not be aware of the language it's generating directly from the output token because of the existence of shared tokens. In other words, the relation between target language and output tokens is weakened because of the shared tokens among different target languages, which might cause representation degradation in the encoder and further lead to off-target in zero-shot test.

3.6 Separating Vocab of Different Languages is Effective yet Expensive

Based on the previous discussion, we now have an idea that maybe we can ease the off-target problem by decreasing the lexical similarity among languages, i.e. decreasing the shared tokens.

When building the vocab for multilingual NMT model, most work regard all languages as one and learn a unified tokenization model. We argue that this leads to low divergence of token distribution since many sub-words are shared across languages.

There is an easy method to decrease the shared tokens without changing the tokenization. We can

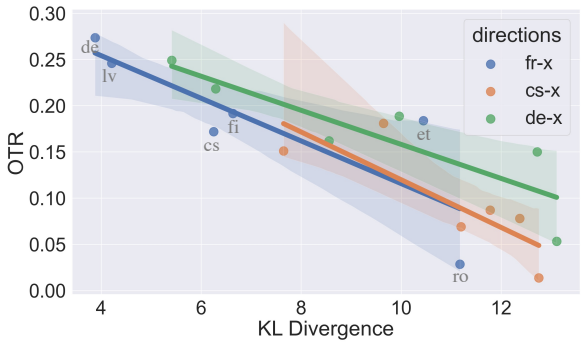


Figure 3: Scatter plot of off-target rate and KL-divergence for different language pairs. We draw the linear regression result with 95% confidence interval.

Method	Size	OTR	BLEU
Vocab Sharing	308M	29%	10.2
Separate Vocab (Dec)	515M	**5%**	**12.4**
Separate Vocab (Enc,Dec)	722M	84%	2.1

Table 2: Average zero-shot result for models with different vocab. (Dec) means only the decoder uses the separate vocab. (Enc,Dec) means both the encoder and the decoder use the separate vocab.

separate the vocab of different languages as shown in Figure 9 from Appendix. Under such condition, no two languages share the same token.

As shown in Table 2, with separate decoder vocab the average off-target rate in 90 directions is reduced from 29% to 5% and the BLEU score is raised from 10.2 to 12.4. We conduct the same probing experiment on encoder representation with the original WMT'10 dataset. As shown in Figure 4, representations for different target are divided. The "chaos" area does not exist anymore.

We also train the model with separated encoder&decoder vocab and finds it suffers from worse zero-shot performance compared to baseline. This also agrees to Rios et al. (2020)'s findings.

We think that without any vocabulary sharing among languages, the model will learn a wrong correlation between input language and output language and ignore the target language identifier during the English-centric training process.

The experiment result justifies our assumption in Section 3.5 that the shared tokens in the decoder will lead to the representation problem. Though achieving great improvement by isolating all vocabulary, it is much more parameter-consuming. In fact, in our experiment, the number of parameters increases from 308M to 515M.

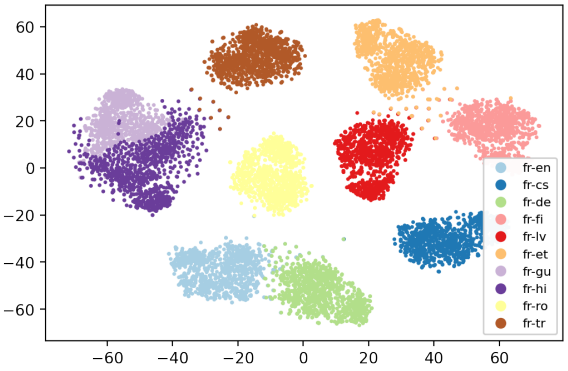


Figure 4: Encoder pooled output visualization using TSNE for French-to-Many translation using separate vocab. The result is comparable to Figure 2, which shows result with shared vocab.

4 Language-Aware Vocabulary Sharing

4.1 Adding Language-Specific Tokens

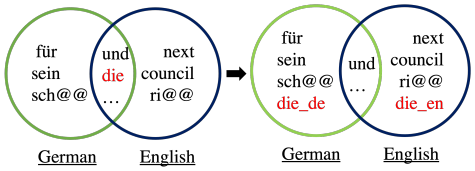


Figure 5: Illustration of LAVS. Tokens with higher shared frequency are split into language-specific ones.

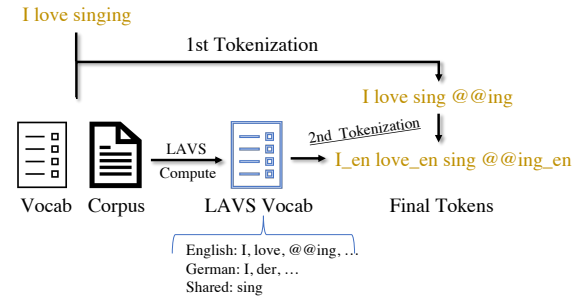
Based on previous observation, lexical similarity will cause the representation degradation problem and further lead to off-target. Thus, our goal is to decrease the lexical similarity. We can achieve it without changing the original tokenizer by splitting the shared tokens into language-specific ones.

As shown in Figure 5, instead of splitting all shared tokens, we can choose specific tokens to

Algorithm 1 Language-Aware Vocabulary Sharing

Input: Shared vocabulary set V', language list L, language's token distributions P and the number of extra language-specific tokens N.
Output: V_{out} is the output vocabulary set.

MaxFreqs = PriorQueue(length=N) ▷ queue that ranks the input elements E from high to low based on E[0].
for i in V' **do**
  for m in L, n in L **do**
    if $m < n$ **then**
      freq = min($P_m^{V'}(i)$,$P_n^{V'}(i)$)
      MaxFreqs.add([freq,m,n,i])
$V_{out} = V'$
for T in MaxFreqs **do**
  m, n, i = T[1], T[2], T[3]
  $V_{out} = V_{out} \cup (V'[i], L[m]) \cup (V'[i], L[n])$
return V_{out}



Input Sentence: I love singing.
Sentencepiece Tokenization: I love sing @@ing .
LAVS Tokenization: I_en love_en sing @@ing_en .
Training/Inference
LAVS Detokenization: I love sing @@ing .
Sentencepiece Detokenization : I love singing.

Figure 6: Illustration of tokenization and detokenization process with Language-Aware Vocabulary Sharing.

split. After decoding, we could simply remove all language-specific tags to restore the literal output sentence. By adding language-specific tokens, the number of shared tokens between different languages decreases and makes the token distribution more different thus increasing the KL Divergence.

4.2 Optimization Goal

Given original vocab set V' and language list L, we aim at creating new vocab V to maximize the average KL divergence within each language pair under the new vocabulary with the restriction of adding N new language-specific tokens. Thus, our objective becomes:

$$V^* = \arg\max_{V} \frac{1}{|L|^2} \sum_{m \in L} \sum_{n \in L} D_{KL}(P_m^V || P_n^V)$$
$$s.t. \quad V' \subseteq V, \quad |V| - |V'| = N \tag{2}$$

where P_m^V denotes the m-th language's token distribution on vocabulary V, add-one smoothing is applied to avoid zero probability. It is a combinatorial optimization problem. The searching space of V has an astronomical size of $C^N_{|V'| \cdot |L|}$.

4.3 Greedy Selection Algorithm that Maximizes Divergence Increment

Based on the previous discussion, we propose the Language-Aware Vocabulary Sharing algorithm as listed in Algorithm 1 to add language-specific tokens. Intuitively, LAVS algorithm prefers to split those shared tokens that have high frequency among different languages, which directly reduces the appearance of shared tokens in the decoder to the maximum extent.

First, we adopt a prior queue to keep the token candidates. Second, for each token in the shared vocabulary, we compute the shared token frequency in each language pair and add the (frequency, languageA, languageB, token) tuple to the queue. Last, since the queue ranks the elements by frequency, we create language-specific tokens for the top N tuples and return the new vocab. We give more details about the algorithm in Appendix B.

The whole tokenization process with LAVS is illustrated in Figure 6. In practice, given an original shared vocab with M tokens, we can always first learn a vocab with $M - N$ tokens and conduct LAVS to add N language-specific tokens to maintain the vocab size M unchanged.

5 Experiments

5.1 Datasets

Following Wang et al. (2020), we collect WMT'10 datasets for training. The devtest split of Flores-101 is used to conduct evaluation. Full information of datasets is in Appendix C.

5.2 Vocabulary Building

Vocab Sharing We adopt Sentencepiece (Kudo and Richardson, 2018) as the tokenization model. We randomly sample 10M examples from the training corpus with a temperature of 5(Arivazhagan et al., 2019) on different directions and learn a shared vocabulary of 64k tokens.

Separate Vocab Based on the sharing vocab of the baseline model, we separate the vocab of each language forming a 266k vocab.

LAVS We first learn a 54k vocabulary using the same method as the baseline model's and add 10k language-specific tokens using LAVS.

5.3 Training Details of MNMT

Architecture We use the Transformer-big model (Vaswani et al., 2017) implemented by fairseq (Ott et al., 2019) with $d_{model} = 1024$, $d_{hidden} = 4096$, $n_{heads} = 16$, $n_{layers} = 6$. We add a target language identifier <XX> at the beginning of input tokens to indicate the translation directions as suggested by Wu et al. (2021).

Optimization We train the models using Adam (Kingma and Ba, 2015), with a total batch

Method	Size	Zero-Shot Off-Target Rate					BLEU Score						
		x-y	H-H	L-L	H-L	L-H	x-y	H-H	L-L	H-L	L-H	en-x	x-en
Vocab Sharing	308M	29%	27%	50%	21%	23%	10.2	11.26	5.03	9.18	9.95	24.8	30.2
Separate Vocab (Dec)	515M	**5%**	4%	19%	**1%**	**1%**	12.4	14.69	6.54	**10.10**	**12.22**	24.6	**30.5**
LAVS (Enc, Dec)	308M	12%	**3%**	33%	13%	6%	**12.5**	**15.90**	6.26	9.91	12.14	24.8	30.3
LAVS (Dec)	308M	8%	13%	**14%**	3%	4%	12.1	13.33	**7.81**	9.80	12.01	**24.9**	30.3

Table 3: Overall performance comparison. x-y denotes all zero-shot directions. H and L denotes High/Low-resources. All evaluation are done with Flores-101 dataset. (Dec) suggests vocab only changes in decoder and (Enc, Dec) suggests changing in both encoder and decoder. LAVS outperforms baseline in zero-shot setting on both BLEU and OTR by a large margin while maintaining the en-x and x-en performance.

Metric	Method	cs-x	fr-x	de-x	fi-x	lv-x	et-x	ro-x	hi-x	tr-x	gu-x
OTR	Vocab Sharing	18.8%	28.3%	22.6%	19.5%	19.2%	17.1%	22.0%	35.2%	30.1%	52.8%
	LAVS(Dec)	**4.2%**	**14.4%**	**11.5%**	**6.2%**	**3.7%**	**4.7%**	**2.9%**	**9.7%**	**10.2%**	**6.1%**
	Δ ↓	-14.6%	-13.9%	-11.1%	-13.3%	-15.5%	-12.4%	-19.1%	-25.5%	-19.9%	-46.7%
BLEU	Vocab Sharing	10.9	10.5	11.3	9.0	9.4	10.0	11.7	6.9	7.3	4.7
	LAVS(Dec)	**12.0**	**12.0**	**12.2**	**9.6**	**10.9**	**11.0**	**14.0**	**9.3**	**9.1**	**8.4**
	Δ ↑	+1.1	+1.5	+0.9	+0.6	+1.5	+1.0	+2.3	+2.4	+1.8	+3.7
BERT Score	Vocab Sharing	0.781	0.808	0.787	0.766	0.783	0.774	0.791	0.771	0.643	0.677
	LAVS(Dec)	**0.799**	**0.829**	**0.806**	**0.786**	**0.790**	**0.798**	**0.796**	**0.777**	**0.660**	**0.713**
	Δ ↑	0.018	0.021	0.019	0.020	0.007	0.024	0.005	0.006	0.017	0.036

Metric	Method	x-cs	x-fr	x-de	x-fi	x-lv	x-et	x-ro	x-hi	x-tr	x-gu
OTR	Vocab Sharing	22.4%	17.8%	23.9%	26.0%	21.9%	28.1%	8.9%	25.4%	14.0%	77.0%
	LAVS(Dec)	**8.7%**	**5.9%**	**6.6%**	**9.2%**	**8.4%**	**7.8%**	**3.0%**	**1.7%**	**7.0%**	**15.4%**
	Δ ↓	-13.7%	-11.9%	-17.3%	-16.8%	-13.5%	-20.3%	-5.9%	-23.7%	-7.0%	-61.6%
BLEU	Vocab Sharing	11.0	17.9	13.2	8.3	12.2	9.9	14.0	8.3	8.8	3.3
	LAVS(Dec)	**12.5**	**20.1**	**15.7**	**9.4**	**13.3**	**11.7**	**14.2**	**9.9**	**9.0**	**6.7**
	Δ ↑	+1.5	+2.2	+2.5	+1.1	+1.1	+1.8	+0.2	+1.6	+0.2	+3.4
BERT Score	Vocab Sharing	0.772	0.776	0.781	0.749	0.757	0.759	0.771	0.743	0.750	0.723
	LAVS(Dec)	**0.791**	**0.799**	**0.796**	**0.770**	**0.777**	**0.774**	**0.797**	**0.756**	**0.768**	**0.726**
	Δ ↑	0.019	0.023	0.015	0.021	0.020	0.015	0.026	0.013	0.018	0.003

Table 4: The zero-shot translation performance (Off-Target Rate, BLEU and BERT-Score) on average x-to-many and many-to-x directions using LAVS (Dec) compared to baseline.

size of 524,288 tokens for 100k steps in all experiments on 8 Tesla V100 GPUs. The sampling temperature, learning rate and warmup steps are set to 5, 3e-4 and 4000.

Back-Translation Back-Translation method is effective in improving zero-shot performance by adding pseudo parallel data generated by the model (Gu et al., 2019; Zhang et al., 2020). For simplicity, we apply off-line back-translation to both the baseline and LAVS. With the trained model, we sample 100k English sentences and translate them to other 10 languages, which creates 100k parallel data for every zero-shot language pair and results in a fully-connected corpus of 9M sentence pairs. We add the generated data to the training set and train the model for another 100k steps.

Evaluation We report detokenized BLEU using sacrebleu[2]. We also report the Off-Target rate with language detector[3] and conduct model-based evaluation using Bert-Score[4] (Zhang* et al., 2020).

5.4 Results

LAVS improves zero-shot translation by a large margin. Table 3 and 4 list the overall results on both zero-shot and supervised directions. According to Table 3, we can see that LAVS improves all the x-to-many and many-to-x directions with a maximum average improvement of -61.6% OTR, +3.7 BLEU and +0.036 Bert-Score compared to the baseline vocab. It gains an average of -21%

[2]nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.1.0
[3]https://github.com/Mimino666/langdetect
[4]https://github.com/Tiiiger/bert_score

Data	OTR	x-y	en-x	x-en	Extra Cost
Vocab Sharing	29%	10.2	24.8	30.2	-
+ B.T.	1%	16.4	23.4	30.0	24 GPU Days
LAVS (Dec)	8%	12.1	**24.9**	30.3	0
+ B.T.	**0%**	**16.8**	23.7	**30.4**	24 GPU Days

Table 5: Results with Back-Translation.

OTR, +1.9 BLEU and +0.02 Bert-Score improvement on 81 zero-shot directions. Compared with the Separate Vocab (Dec) method which also leads to significant improvement in x-y directions, LAVS does not increase any model size.

LAVS with Back-Translation further improves the zero-shot performance. As shown in Table 5, as expected, our back-translation method can improve the zero-shot performance by a large margin. Under such setting, LAVS also outperforms Vocab Sharing by 0.4 average BLEU score on zero-shot directions.

We also observe performance degradation in English-to-Many directions for both models comparing to not using back-translation, which also agrees to the result of Zhang et al. (2020); Rios et al. (2020). We think a possible reason is that the English-to-Many performance will be interfered with the increase of translation tasks. Back Translation also brings much extra cost. The total training time for the model with Back-Translation is almost twice as long as the model with vanilla training. Only applying LAVS brings no extra training cost and does not influence the supervised performance.

6 Discussion

6.1 How does LAVS calibrate the direction?

We visualize the encoder-pooled representations for model with LAVS(dec) in Figure 7. The representations' distribution is similar to Figure 4 where representations for different target are almost divided, suggesting that LAVS work similarly to separating all the vocabulary for different languages. We also give a case study as shown in Section 6.2.

We further visualize the language identifiers' hidden output during among high-resource languages and compare the results of the original Vocabulary Sharing and LAVS. As shown in Figure 10 from Appendix, it turns out that LAVS encodes more discriminative target language information into the <XX> token's hidden output.

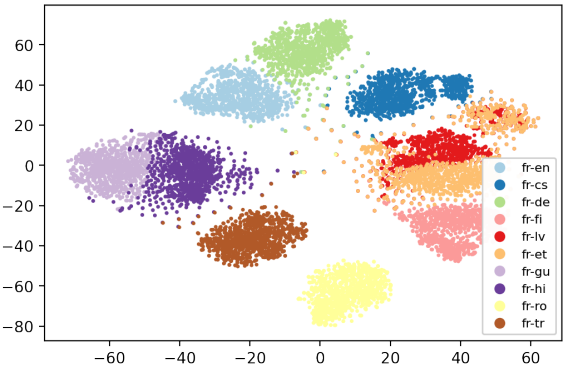


Figure 7: The encoder-pooled representations learned by multilingual NMT with LAVS on fr-x directions.

6.2 Case Study

We compare different model's outputs as shown in Figure 8. The baseline output has off-target problem while LAVS output generates in the correct language. From the direct token output of LAVS, we can see that many of which are language-specific tokens. Models with LAVS could learn the relation between the target language signal and corresponding language-specific tokens, which further decreases the probability of off-target.

Direction: DE-> FR
Input: <FR> Apia wurde in den 50ern des 18. Jahrunderts gegründet und ist seit 1959 die offizielle Hauptstadt von Samoa.
Output(baseline): Apia was founded in the 50s of the 18th century and is the official capital of Samoa since 1959. (Off-Target to English)
Gold: Apia a été fondée dans les années 1850 et est la capitale officielle des Samoa depuis 1959.

Output(LAVS-token): Apia_fr a_fr été fondée dans les_fr 50 ans_fr du_fr 18e siècle et_fr est_fr depuis 1959 la_fr capitale officielle de_fr Samoa.
Output(LAVS-literal): Apia a été fondée dans les 50 ans du 18e siècle et est depuis 1959 la capitale officielle de Samoa.

Figure 8: Case study of DE->FR zero-shot translation. The baseline model off-target to English. Tokens in blue belong to language-specific tokens.

6.3 Scalability of LAVS

As shown in Table 6, we explore how the number of language specific(LS) tokens influence the zero-shot performance. The result shows that the OTR keeps decreasing when the number of LS tokens increases. It suggests that more LS tokens can

Shared Tokens(M)	LS Tokens(N)	OTR↓	Sup. BLEU↑
64k	0	29.4%	27.5
54k	0	33.1%	26.9
54k	10k	8.2%	27.6
54k	20k	7.4%	**27.8**
54k	50k	5.9%	27.6
54k	212k	**5%**	27.6

Table 6: Exploration in number of Language-Specific tokens in LAVS(dec) and the Off-Target Rate on Flores-101. We report the average OTR on zero-shot directions and average BLEU on supervised directions.

Data	OTR↓	x-y↑	en-x↑	x-en↑
Vocab Sharing	72%	1.9	12.6	19.8
LAVS (Dec)	**58%**	**2.3**	**13.5**	**20.1**

Table 7: Results in OPUS dataset. We evaluate 1722 zero-shot directions and 84 supervised-directions.

better relieve the off-target issue **without harming the supervised performance.**

To test how LAVS generalizes in dataset with more languages, we compare LAVS and VS on OPUS-100 (Zhang et al., 2020). More details of the experiment can be found in Appendix D To alleviate the inference burden, we select all 42 languages with 1M training data for evaluation, which results in 1722 zero-shot directions and 84 supervised directions (en-x and x-en). As shown in Table 7, it turns out that LAVS can improve the zero-shot performance(-14% OTR, detailed results in Table 12 from appendix) under such setting. Yet, the overall performance is much lower comparing to training on WMT'10. With more languages, the lack of supervision signal would become more problematic for zero-shot translation. LAVS improves the en-x performance by a large margin (+0.9 BLEU, detailed scores in Table 13 from appendix), we think separate the vocab of different languages on decoder might have positive influence on general en-x performance.

6.4 LAVS's Compatibility with Masked Constrained Decoding

We propose another method to prevent off-target, which is through masked constrained decoding (MCD). During decoding, the decoder only considers tokens that belong to the target vocab in softmax. The target vocab could be computed using the training corpus. We implement MCD for both original vocab sharing and LAVS. We list the detail of the size of different target vocabs in Table 11 from appendix.

Method	DE->CS		FR->DE	
	OTR	BLEU	OTR	BLEU
Vocab Sharing	45.1%	9.7	38.3%	12.7
w/ MCD	30.9%	11.4	36.4%	12.8
LAVS (Dec)	18.9%	13.0	15.4%	17.2
w/ MCD	**11.1%**	**14.2**	**11.3%**	**17.8**

Table 8: The results of masked constrained decoding (MCD) combined with LAVS. Constrained decoding could further improve the performance of LAVS.

As shown in Table 8, it turns out that the method can further improve the zero-shot performance for LAVS (+1.2 BLEU for de-cs, +0.6 BLEU for fr-de). It is worth noticing that, in some direction like FR->DE, the benefit of MCD is rather small for the baseline model (+0.1 BLEU). We think the reason is that the original vocab sharing generates many shared tokens between languages, which will weaken the influence of the constraint. Thus, with more language-specific tokens, LAVS can work better with constrained decoding.

7 Conclusion

In this paper, we delve into the hidden reason for the off-target problem in zero-shot multilingual NMT and propose Language-Aware Vocabulary Sharing (LAVS) which could significantly alleviate the off-target problem without extra parameters. Our experiments justify that LAVS creates a better multilingual vocab than the original Vocabulary Sharing method for multiple languages.

8 Limitation

LAVS is proposed to overcome the off-target problem among languages that share alphabets because those languages tend to have more sharing tokens after the sub-word tokenization process. As for language pair that does not have shared tokens, LAVS might not have a direct influence on the zero-shot translation though it can also increase the overall performance for those languages, which might need further exploration.

9 Acknowledgements

This paper is supported by the National Key Research and Development Program of China under Grant No.2020AAA0106700 and the National Science Foundation of China under Grant

No.61936012. We also thank all reviewers for their valuable suggestions.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, and Omar Zaidan, editors. 2010. *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Association for Computational Linguistics, Uppsala, Sweden.

Liang Chen, Runxin Xu, and Baobao Chang. 2022. Focus on the target's vocabulary: Masked label smoothing for machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 665–671, Dublin, Ireland. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

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

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.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english.

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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

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

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *ArXiv*, abs/2106.13736.

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 NAACL-HLT 2019: Demonstrations*.

Annette Rios, Mathias M"uller, 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.

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

Nllb team, Marta Ruiz Costa-jussà, James Cross, Onur cCelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Alison Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon L. Spruit, C. Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao,

Vedanuj Goswami, Francisco Guzm'an, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *ArXiv*, abs/2207.04672.

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 *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020. Multi-task learning for multilingual neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1022–1034, 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.

Yonghui Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason R. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Jingjing Xu, Hao Zhou, Chun Gan, Zaixiang Zheng, and Lei Li. 2021a. Vocabulary learning via optimal transport for 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 7361–7373, Online. Association for Computational Linguistics.

Weijia Xu, Yuwei Yin, Shuming Ma, Dongdong Zhang, and Haoyang Huang. 2021b. Improving multilingual neural machine translation with auxiliary source languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3029–3041, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. 2021. 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, 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.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Method for Completely Separating Vocab

It is easy to turn a shared vocabulary into a separate vocabulary for different languages. As shown in Figure 9, we can split the shared token into language specific token if it appears in more than one language.

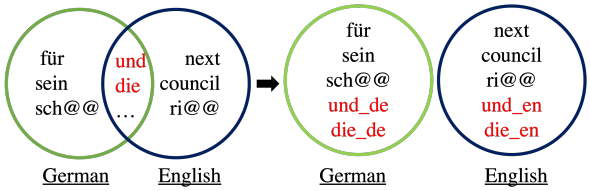


Figure 9: Illustration of completely separating vocabulary of different languages. Note that we don't need to learn a new vocab. Given the original shared vocab, we can split those tokens that are shared by two or more languages into language-specific ones and get a fully separate vocab for each language.

B Separating Tokens by Frequency

We can also view LAVS from the optimization goal's perspective. We start from only two languages J and Q and compute KL-divergence's change if we only split one shared token to two language-specific tokens.

$$\begin{aligned}\Delta D^{i}_{KL} &= -J(i)log\frac{J(i)}{Q(i)} - Q(i)log\frac{Q(i)}{J(i)} + \lambda \\ &= [J(i) - Q(i)]log\frac{Q(i)}{J(i)} + \lambda\end{aligned} \tag{3}$$

where we will have two i-th tokens for the different languages from the original vocabulary. λ is the smoothing factor that can be seen as a constant. According to equation 3, splitting token that has more similar occurrence probability in the two languages will lead to higher increment in language's KL-Divergence (If $J(i)! = Q(i)$, either the $J(i) - Q(i)$ term or the log term will be negative, and the multiply result will also be negative. If $J(i) = Q(i)$ it will be zero, thus reaching the maximum). Also considering the fact that the tokens with high frequency influence the training process much more than the near-zero ones, we should first split the tokens that appear in *two or more* languages all with *high frequency*.

C Datasets

C.1 WMT'10

Following Wang et al. (2020); Yang et al. (2021); Xu et al. (2021b), we collect data from freely-accessible WMT contests to form a English-Centric WMT10 dataset.

Direction	Train	Test	Dev
Fr↔En	10.00M	newstest15	newstest13
Cs↔En	10.00M	newstest18	newstest16
De↔En	4.60M	newstest18	newstest16
Fi↔En	4.80M	newstest18	newstest16
Lv↔En	1.40M	newstest17	newsdev17
Et↔En	0.70M	newstest18	newsdev18
Ro↔En	0.50M	newstest16	newsdev16
Hi↔En	0.26M	newstest14	newsdev14
Tr↔En	0.18M	newstest18	newstest16
Gu↔En	0.08M	newstest19	newsdev19

Table 9: Description for WMT'10 Dataset.

C.2 Flores-101

Flores-101 (Goyal et al., 2021; Guzmán et al., 2019) is a Many-to-Many multilingual translation benchmark dataset for 101 languages. It provides parallel corpus for all languages, which makes it suitable to test the zero-shot translation performance of multilingual NMT model. We use the devtest split of the dataset, and only test on the languages that appear during supervised training.

Language	Code	Split	Size
French	Fr	devtest	1012
Czech	Cs	devtest	1012
German	De	devtest	1012
Finnish	Fi	devtest	1012
Latvian	Lv	devtest	1012
Estonian	Et	devtest	1012
Romanian	Ro	devtest	1012
Hindi	Hi	devtest	1012
Turkish	Tr	devtest	1012
Gujarati	Gu	devtest	1012

Table 10: Description for Flores-101 Dataset.

D Experiment on OPUS-100 dataset

OPUS-100(Zhang et al., 2020) is an English-centric dataset consisting of parallel data between

English and 100 other languages. We removed 5 languages (An, Dz, Hy, Mn, Yo) from OPUS, since they are not paired with a dev or testset and train the models with all remaining data. The training configuration is the same as our experiment on WMT'10 dataset. The baseline vocab size is 64k. We also implement the baseline model with a larger vocab (256k) but the performance is much lower than the 64k version so we keep the vocab size to 64k. For LAVS, We set the number of language-specific token to 150k instead of 10k because of the increase of languages. We evaluate the supervised and zero-shot performance on Flores-101 dataset. To alleviate the inference burden, we select all 42 languages with 1M training data to conduct zero-shot translation, which forms 1722 zero-shot directions at all. The ISO code of the evaluated lanugages are "ar, bg, bn, bs, ca, cs, da, de, el, es, et, fa, fi, fr, he, hr, hu, id, is, it, ja, ko, lt, lv, mk, ms, mt, nl, no, pl, pt, ro, ru, sk, sl, sr, sv, th, tr, uk, vi, zh".

E Visualize the language identifiers' representation

During zero-shot translation, the language identifier token "<XX>" is the only element indicating the correct direction. Similar to the visualization in Section 3.3, as shown in Figure 10, we visualize the <XX> tokens' hidden output(instead of the pooled result from all input tokens) during French-to-Many translation among high-resource languages and compare the results of the original Vocabulary Sharing and LAVS. It turns out that LAVS encodes more discriminative target language information into the <XX> token's hidden output, while the original Vocabulary Sharing fails on that.

In original Vocabulary Sharing the mapping between the target language identifier <XX> and output token is Many-to-One since different language could share output tokens. While for LAVS, the mapping becomes One-to-One for a part of tokens, impulsing the encoder to learn more discriminative representations for the target language identifier.

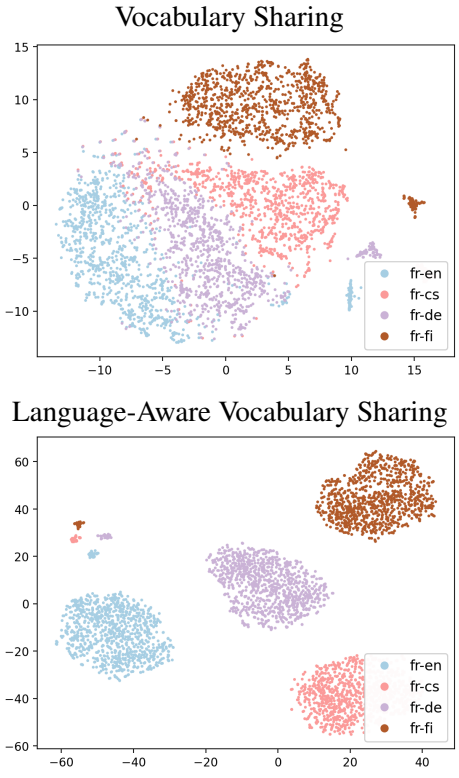


Figure 10: Encoder's hidden output for language identifier token <XX>, visualized using TSNE.

Language	Code	LAVS	VS
French	Fr	28k	33k
Czech	Cs	25k	30k
German	De	29k	35k
Finnish	Fi	23k	28k
Latvian	Lv	24k	29k
Estonian	Et	15k	18k
Romanian	Ro	14k	20k
Hindi	Hi	10k	11k
Turkish	Tr	11k	12k
Gujarati	Gu	7k	9k

Table 11: Size of different target vocab for LAVS and VS vocab. Both vocabs have 64k tokens at all. Original VS generally has more tokens in each target vocab, which would weaken the effect of the constrain mask.

	ar	bg	bn	bs	ca	cs	da	de	el	es	et	fa	fi	fr	he	hr	hu	id	is	it	ja	ko	lt	lv	mk	ms	mt	nl	no	pl	pt	ro	ru	sk	sl	sr	sv	th	tr	uk	vi	zh	AVG
VS	0.96	0.87	0.39	0.56	0.53	0.92	0.61	0.92	**0.91**	0.80	0.73	0.61	0.86	0.76	0.69	0.58	0.89	0.42	0.85	0.83	0.58	**0.64**	0.81	0.65	0.78	0.43	**0.43**	0.87	0.64	0.91	0.71	0.87	0.85	0.85	0.78	0.85	0.75	**0.63**	0.48	0.83	0.47	**0.72**	0.72
LAVS	**0.93**	**0.67**	**0.30**	**0.35**	**0.51**	**0.66**	**0.48**	**0.76**	0.96	**0.73**	**0.49**	**0.47**	**0.63**	**0.69**	**0.68**	**0.37**	**0.78**	**0.27**	**0.80**	**0.74**	**0.51**	0.68	**0.50**	**0.49**	**0.57**	**0.27**	**0.43**	**0.64**	**0.50**	**0.75**	**0.63**	**0.74**	**0.60**	**0.54**	**0.52**	**0.66**	**0.57**	0.74	**0.21**	**0.67**	**0.47**	0.77	**0.58**

Table 12: Detailed zero-shot OTR of X-to-Many experiment on OPUS-100. Each score denotes the average OTR from X to other 41 languages.

	ar	bg	bn	bs	ca	cs	da	de	el	es	et	fa	fi	fr	he	hr	hu	id	is	it	ja	ko	lt	lv	mk	ms	mt	nl	no	pl	pt	ro	ru	sk	sl	sr	sv	th	tr	uk	vi	zh	AVG
VS	7.3	17.5	6.4	12	24.7	12.7	25.5	16.5	9.7	17.2	10.2	4.2	7.4	27.7	6.8	**12.2**	8.6	20.2	5.3	15.8	2.0	1.6	11.1	13.7	17.2	18.5	21.1	14	**19.9**	7.5	26.1	15.8	12	13.6	12	0.3	22.3	3	6.3	5.6	13.3	6.7	12.6
LAVS	**7.7**	**18.7**	**6.7**	**13.6**	**25.7**	**12.9**	**26.2**	**18.5**	**11.0**	**18.0**	**10.9**	**4.6**	**8.3**	**29.1**	**7.4**	12.0	**9.4**	**21.9**	**6.2**	**17.4**	**2.5**	**2.3**	**12.0**	**14.1**	**17.5**	**20.5**	**22.0**	**14.7**	17.5	**8.1**	**27.2**	**16.2**	**12.5**	**14.6**	**12.7**	**0.5**	**23.5**	**3.7**	**7.4**	**8.0**	**15.3**	**7.5**	**13.5**

Table 13: Detailed BLEU scores of English-to-Many experiment on OPUS-100.

ACL 2023 Responsible NLP Checklist

A For every submission:

☐ A1. Did you describe the limitations of your work?
Left blank.

☐ A2. Did you discuss any potential risks of your work?
Left blank.

☐ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

☐ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
Left blank.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

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

C ☐ Did you run computational experiments?

Left blank.

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

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

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

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

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

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

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.