Tencent AI Lab - Shanghai Jiao Tong University Low-Resource Translation System for the WMT22 Translation Task

Zhiwei He[*]
Shanghai Jiao Tong University
zwhe.cs@sjtu.edu.cn

Xing Wang[†]
Tencent AI Lab
brightxwang@tencent.com

Zhaopeng Tu
Tencent AI Lab
zptu@tencent.com

Shuming Shi
Tencent AI Lab
shumingshi@tencent.com

Rui Wang
Shanghai Jiao Tong University
wangrui12@sjtu.edu.cn

Abstract

This paper describes Tencent AI Lab - Shanghai Jiao Tong University (TAL-SJTU) Low-Resource Translation systems for the WMT22 shared task. We participate in the general translation task on English⇔Livonian. Our system is based on M2M100 (Fan et al., 2021) with novel techniques that adapt it to the target language pair. (1) Cross-model word embedding alignment: inspired by cross-lingual word embedding alignment, we successfully transfer a pre-trained word embedding to M2M100, enabling it to support Livonian. (2) Gradual adaptation strategy: we exploit Estonian and Latvian as auxiliary languages for many-to-many translation training and then adapt to English-Livonian. (3) Data augmentation: to enlarge the parallel data for English-Livonian, we construct pseudo-parallel data with Estonian and Latvian as pivot languages. (4) Fine-tuning: to make the most of all available data, we fine-tune the model with the validation set and online back-translation, further boosting the performance. In model evaluation: (1) We find that previous work (Rikters et al., 2022) underestimated the translation performance of Livonian due to inconsistent Unicode normalization, which may cause a discrepancy of up to 14.9 BLEU score. (2) In addition to the standard validation set, we also employ round-trip BLEU to evaluate the models, which we find more appropriate for this task. Finally, our unconstrained system achieves BLEU scores of 17.0 and 30.4 for English to/from Livonian.[1]

1 Introduction

This paper introduces our submissions to the WMT22 general machine translation task. Last year, Tencent AI Lab participated in two translation tasks: News (Wang et al., 2021a) and Biomedical translation (Wang et al., 2021b). This year, we participate in English⇔Livonian (En⇔Liv), a very low-resource and distant language pair. Considering the scarcity of parallel En-Liv corpus, we only participate in the unconstrained evaluation.

We use M2M100 1.2B[2] (Fan et al., 2021) as the pre-trained model which is a massive multilingual translation model that supports any pair of 100 languages[3] and shows promising performance for low-resource translation. To adapt it to En-Liv, the first thing to do is enabling it to support Liv. A common approach is to expand the vocabulary and the word embedding matrix to contain the extra tokens. However, the incoming embeddings must be randomly initialized (Garcia et al., 2021; Bapna et al., 2022), which leads to inconsistency with the original embeddings and increases training difficulty. Fortunately, Rikters et al. (2022) has released a translation model for En-Liv called Liv4ever-MT[4]. Inspired by supervised cross-lingual word embedding alignment (Lample et al., 2018b), we propose cross-model word embedding alignment (CMEA) that learns a linear transformation between the embedding matrices of two models. Therefore, the incoming embeddings can be extracted from Liv4ever-MT and transformed to M2M100's word embedding space rather than random initialization.

In terms of model training, we adopt a gradual adaptation strategy. The overall training process is shown in Figure 1. Following Rikters et al. (2022), we also use Estonian (Et) and Latvian (Lv) as auxiliary languages. Liv has been influenced by Et and Lv for centuries. There are about 800 Et loanwords and 2,000 Lv loanwords in Liv (Décsy, 1965). Therefore, we first add Et and Lv for many-to-many translation training, resulting in a 4-lingual

[*] Work was done when Zhiwei He was interning at Tencent AI Lab.

[†] Xing Wang is the corresponding author.

[1] Code, data, and trained models are available at `https://github.com/zwhe99/WMT22-En-Liv`.

[2] `https://github.com/facebookresearch/fairseq/tree/main/examples/m2m_100`

[3] M2M100 supports English, Latvian and Estonian.

[4] `https://huggingface.co/tartuNLP/liv4ever-mt`

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 260–267
Abu Dhabi, December 7–8, 2022. ©2022 Association for Computational Linguistics

translation model. We then augment the En-Liv data with forward and backward translations using Et and Lv as the pivot languages. Finally, we combine all the authentic and synthetic data to retrain the model, followed by a few steps of fine-tuning with the validation set and online back-translation.

In terms of model evaluation, we find that the data set provided by Rikters et al. (2022) suffers from inconsistent Unicode normalization. This inconsistency is reflected in using two or more encodings for the same character, which leads to inconsistent encoding between model hypothesis and reference[5] and thus inaccurate evaluation. In our experiments, normalizing the character encoding can bring an average improvement of +2.5 BLEU on the liv4ever[6] test set (see appendix A) and up to +14.9 BLEU on a subset from a specific source. In addition to the standard validation set, we also employ round-trip BLEU to evaluate our models, which is an effective unsupervised criterion (Lample et al., 2018a) and reduces the demand for the parallel corpus. Zhuo et al. (2022) have found that in the scope of neural machine translation, round-trip translation quality correlates consistently with forward translation quality. We consider round-trip BLEU a better evaluation method for this task. The reasons for this are threefold: more data, more general domain, and the same original language as the WMT22 En-Liv test set.

This paper is structured as follows: Section 2 describes the data statistics and processing methods. Then we present our evaluation methods in Section 3. Our translation system and ablation study are detailed in Section 4, followed by the final results. Finally, we conclude the paper in Section 5.

2 Data and Processing

2.1 Overview

Statistics Table 1 lists statistics of the parallel and monolingual data we used. We collect parallel data for any pair in {En, Liv, Et, Lv} and collect monolingual data for En and Liv.

Data Source The parallel data is mainly all available corpora from OPUS[7]. Due to the scarcity of data, we include liv4ever-dev in training data and use liv4ever-test as the validation set. For En-Et and En-Lv, we augment them with the parallel data from WMT18 and WMT17, respectively. For En-Liv, En-Lv and Liv-Lv, we collected additional parallel data from Facebook posts of the Livonian Institute and Livones.net[8]. The monolingual En is News Crawl 2007-2021. The monolingual Liv combines all Liv from parallel data and monolingual data from liv4ever[6].

Data	Lang	# Sent.	
		Raw	Filter
Parallel Data	En-Liv	1.2K	1.1K
	En-Et	40.3M	20.7M
	En-Lv	27.2M	11.3M
	Liv-Et	14.8K	14.8K
	Liv-Lv	12.4K	12.2K
	Et-Lv	10.7M	7.0M
Monolingual Data	En	325.6M	281.3M
	Liv	138.2K	50.2K

Table 1: Statistics of parallel and monolingual data. We report the number of sentences before and after filtering.

2.2 Pre-processing

To obtain higher quality training data, we employ a series of data cleaning using Moses toolkit[9] and our scripts[10]. We process parallel data as follows:

- Replace Unicode punctuation, normalize punctuation and remove non-printing characters
- Language identification and filtering
- Remove instances with too much punctuation
- Remove instances with identical source and target sentences
- Remove instances containing URLs
- Remove instances appearing in evaluation data
- Remove instances with more than 175 tokens or length ratio over 1.5

The liv4ever corpus has a small amount of data, and the existing tools may not support Liv well. Therefore, for the liv4ever corpus, we don't apply punctuation processing or language and length ratio filtering. For the monolingual data, we use the same cleaning steps as parallel data except for

[5] SentencePiece does uniform normalization by default. Therefore, the character encoding in the model hypothesis is uniform but may not be consistent with the reference.

[6] https://opus.nlpl.eu/liv4ever-v1.php

[7] https://opus.nlpl.eu/

[8] The numbers of additional sentences collected from Facebook are En-Liv: 54, En-Lv: 61 and Liv-Lv: 61.

[9] https://github.com/moses-smt/mosesdecoder

[10] https://github.com/zwhe99/corpus-tools

identical source-target filtering and length ratio filtering.

After cleaning the data, we apply SentencePiece[11] encoding using the trained model from Liv4ever-MT[4]. We also reuse their vocabulary that shared by all languages.

2.3 Evaluation Data

We regard the liv4ever-test as the validation set, which is a multi-way data set for {En, Liv, Et, Lv} containing 855 unique sentences. Besides, for En⇔Liv evaluation, we collect monolingual English from the source of WMT22 English-German (En-De) test set to compute round-trip BLEU (En⇒Liv⇒En).

3 Model Evaluation

This section describes our methods for model evaluation. Specifically, we explain the Unicode inconsistency problem in the liv4ever data set and the resulting underestimation of model performance. In addition, we introduce round-trip BLEU as the more appropriate way for this competition.

3.1 Unicode inconsistency problem

Rikters et al. (2022) collected the liv4ever data set and built Liv4ever-MT, the first machine translation model for Livonian. We find that the liv4ever data set does not use consistent Unicode normalization, resulting in different encodings for the same character. This did not lead to any training problem in Rikters et al. (2022) because SentencePiece does NFKC[12] normalization by default. However, when computing SacreBLEU[13], the encoding of model output and the reference will be inconsistent, resulting in inaccurate evaluation.

We re-evaluate the performance of Liv4ever-MT before and after normalizing the encoding of references to NFKC. Table 2 shows the SacreBLEU results[14] on the entire test set and a subset from Satversme. Before normalization, our results are very close to those reported in Rikters et al. (2022), while after normalization, the BLEU score improves considerably. In particular, the difference in BLEU score is up to 14.9 on the Lv⇒Liv of the Satversme subset. Therefore, we report SacreBLEU after normalization in the following.

	En-Liv		**Et-Liv**		**Lv-Liv**	
	⇒	⇐	⇒	⇐	⇒	⇐
All						
Liv4ever-MT (Rikters et al.)	11.0	19.0	16.5	23.1	17.7	25.2
Our Eval.	10.9	18.9	16.6	22.9	17.7	24.9
+ Norm. Ref.	**14.3**	**19.3**	**20.5**	**24.4**	**22.3**	**29.3**
Subset (Satversme)						
Liv4ever-MT (Rikters et al.)	7.7	24.5	-	-	-	-
Our Eval.	7.6	24.7	7.2	18.7	9.2	19.4
+ Norm. Ref.	**18.2**	**25.8**	**19.9**	**23.7**	**24.2**	**33.6**

Table 2: BLEU scores of Liv4ever-MT on liv4ever-test. **Liv4ever-MT** (Rikters et al.): copied from Rikters et al. (2022). **Our Eval.**: We use the released Liv4ever-MT to generate translation outputs and re-evaluate them with the original references, which shows similar results compared with Rikters et al. (2022). **+ Norm. Ref.**: re-evaluation after normalizing the encoding of references to NFKC. See Appendix A for all language pairs.

3.2 Round-trip BLEU

We collect monolingual English from the source of WMT22 English-German (En-De) test set and conduct two steps translation: En⇒Liv⇒En. The round-trip BLEU score can be obtained by comparing the original input with the model output English. We regard it a better way to evaluate En⇔Liv performance for this task considering three aspects: (1) En-De test set has 20683 sentences, much more than the liv4ever-test. (2) It may contain more general domain data, while the liv4ever-test is relatively restricted due to the low-resource limitation. (3) The original language used in computing the round-trip BLEU is the same as the WMT22 En-Liv test set (both English-original).

4 System and Ablation Study

In this section, we describe our system in this competition and provide a comprehensive ablation study of the key components.

4.1 System Overview

We depict the overview of our system in Figure 1, which can be divided into five steps:

1. **Cross-model word embedding alignment**: transfer the word embeddings of Liv4ever-MT to M2M100, enabling it to support Livonian.

[11] https://github.com/google/sentencepiece

[12] https://unicode.org/reports/tr15/

[13] https://github.com/mjpost/sacrebleu

[14] nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0

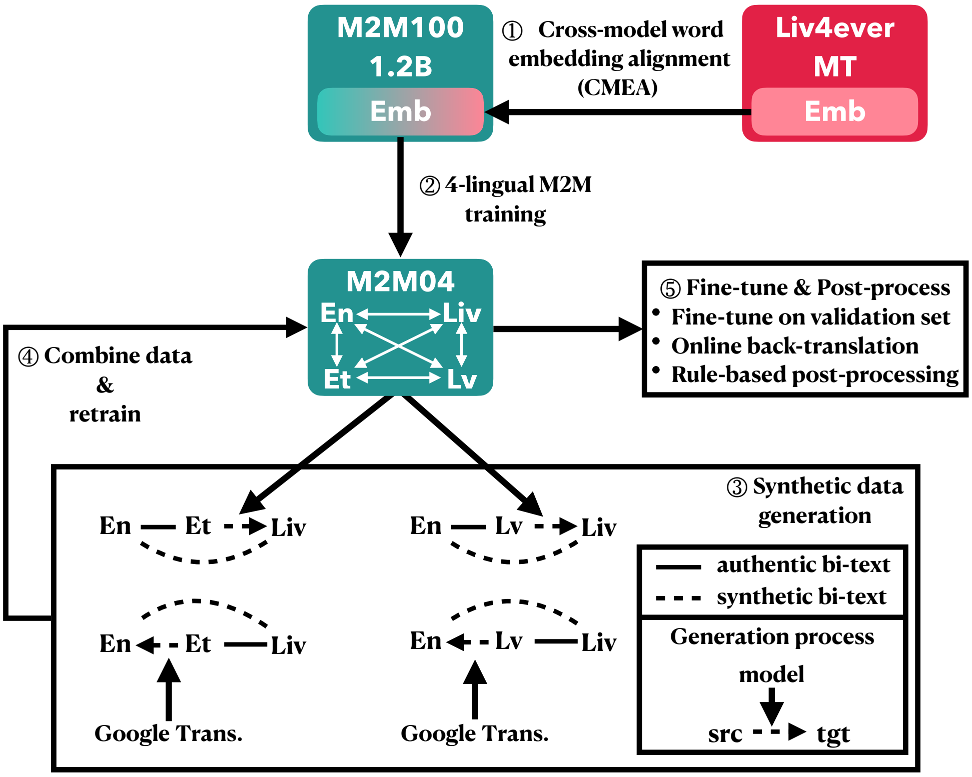

Figure 1: The training process of our translation system.

2. **4-lingual M2M training**: many-to-many translation training for all language pairs in {En, Liv, Et, Lv}, using only parallel data.
3. **Synthetic data generation**: generate synthetic bi-text for En-Liv, using Et and Lv as pivot languages.
4. **Combine data and retrain**: combine all the authentic and synthetic bi-text and retrain the model following step 2.
5. **Fine-tune & post-process**: fine-tune the model on En⇔Liv using the validation set and perform online back-translation using monolingual data. Finally, apply rule-based post-processing to the model output.

4.2 Cross-model Word Embedding Alignment

M2M100 1.2B does not support Livonian. Therefore, we used Liv4ever-MT's SentencePiece model and vocabulary to process all the data. For M2M100, the embeddings of new coming words can be randomly initialized. However, randomly initialized word embeddings and the pre-trained models may not be compatible. Inspired by supervised cross-lingual word embedding alignment (Lample et al., 2018b), we propose cross-model word embedding alignment (CMEA) to transform the trained word embeddings of Liv4ever-MT into M2M100, avoiding random initialization.

CMEA We denote Liv4ever-MT and M2M100 model by l and m. Their corresponding vocabularies and embedding matrices are d_l, d_m and $\mathbf{X}^l$, $\mathbf{X}^m$. Table 3 shows the statistics of the vocabularies. Let $\mathbf{X}^f$ be the final embedding matrix we expected. We adopt d_l as the final vocabulary, which can be divided into two parts:

$\lvert d_l \rvert$	$\lvert d_m \rvert$	$\lvert d_l \cap d_m \rvert$	$\lvert d_l \cap d_m \rvert / \lvert d_l \rvert$
47972	128108	11410	23.8%

Table 3: Statistics of Liv4ever-MT (d_l) and M2M100 (d_m) vocabularies.

$$d_l = (d_l \cap d_m) \cup (d_l - d_m). \tag{1}$$

For the overlapped part $d_l \cap d_m$, $\mathbf{X}^f$ can reuse the embedding from $\mathbf{X}^m$:

$$\mathbf{X}^f_{d_l \cap d_m} = \mathbf{X}^m_{d_l \cap d_m}. \tag{2}$$

For the rest part $d_l - d_m$, we first find a liner transformation $\mathbf{W}$ between two embedding spaces such that:

$$\mathbf{W}^* = \underset{\mathbf{W}}{\arg\min} \|\mathbf{W}\mathbf{X}^l_{d_l \cap d_m} - \mathbf{X}^m_{d_l \cap d_m}\|_F \\ \text{s.t.}\ \mathbf{W}^T\mathbf{W} = \mathbf{I}. \tag{3}$$

According to Everson (1998),

$$\mathbf{W}^* = \mathbf{U}\mathbf{V}^T, \\ \text{with } \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathrm{SVD}\left(\mathbf{X}^m_{d_l \cap d_m}{\mathbf{X}^l_{d_l \cap d_m}}^T\right). \tag{4}$$

Then the word embeddings can be initialized as:

$$\mathbf{X}^f_{d_l - d_m} = \mathbf{W}^*\mathbf{X}^l_{d_l - d_m}. \tag{5}$$

Experiment To investigate the effect of CMEA, we conducted **4-lingual M2M training** with different sampling temperature (Aharoni et al., 2019; Tang et al., 2021). Table 4 shows the BLEU scores on the validation set. We have the following observations:

- M2M04 outperforms Liv4ever-MT by a large margin owing to the larger model size, more training data and the pre-trained parameters.
- On most language pairs, our proposed CMEA initialization significantly improves translation performance compared to random initialization of new coming embeddings.
- Temperature set to 5 with CMEA initialization achieves the best overall results. Therefore, we used this model in **synthetic data generation**.

	En-Liv		**Et-Liv**		**Lv-Liv**	
	⇒	⇐	⇒	⇐	⇒	⇐
Liv4ever-MT Rikters et al.	14.3	19.3	20.5	24.4	22.3	29.3
M2M04 (T=5)	21.1	27.7	25.3	29.2	26.8	36.6
+ CMEA	**23.0**	**28.4**	**27.2**	**30.7**	**28.5**	**37.6**
M2M04 (T=10)	**21.3**	26.6	25.5	27.7	26.3	34.6
+ CMEA	21.1	**27.1**	**26.0**	**29.6**	**27.5**	**36.3**
M2M04 (T=20)	21.9	26.7	**26.5**	**29.8**	27.3	**36.5**
+ CMEA	**22.1**	**27.4**	25.8	27.9	**27.9**	33.8

Table 4: Experimental results of 4-lingual M2M training. We denote M2M04 as the 4-lingual translation model. 'T' represents the sampling temperature.

4.3 Synthetic Data Generation

Data agumentation (Sennrich et al., 2016; Jiao et al., 2020, 2022, 2021; He et al., 2022) is a widely used technique to boost the performance of neural machine translation. To augment the parallel data for En-Liv, we adopt both forward and backward translation to generate synthetic bi-text for En-Liv. Figure 1 (below) illustrates the process of synthetic data generation.

Considering the performances of Et/Lv⇒Liv are much better than En⇒Liv (see Table 4), we use Et and Lv as pivot languages to generate Liv instead of directly generating from En. Taking Et as the pivot language, given authentic En-Et bi-text, we use the best model in Table 4 to translate the Et into Liv, thus forming the synthetic En-Liv which is En-original. Conversely, given authentic Et-Liv, we translate Et into En using Google Translate, forming the synthetic En-Liv which is Liv-original. For Lv as the pivot language, we repeat the same steps. Table 5 lists statistics of the synthetic En-Liv data after filtering.

Data Type	**Pivot Language**	
	Et	Lv
En-original	20.5M	11.2M
Liv-original	14.2K	11.6K

Table 5: The number of sentences of generated synthetic data after filtering, which is divided into four categories based on the original language and the pivot language.

Experiment We combine the authentic and synthetic bi-text and retrain the 4-lingual model. The sampling temperature is set to 0 here to avoid downsampling for En-Liv. When using only En-original or Liv-original synthetic data, we control the sampling frequency of the different language pairs to be consistent with using the full data. Table 6 shows the BLEU scores on the multi-way validation set. We also report the round-trip BLEU on the monolingual En from the source of WMT22 En-De test set, which is En-original. Unexpectedly, original-language greatly affects the model performance and causes inconsistent results between different evaluation methods:

- En-original synthetic data remarkably degrades model performance on the validation set but significantly increases the round-trip BLEU.
- Liv-original synthetic data slightly reduces the performance on the validation set but moderately increases the round-trip BLEU.

- When using both kinds of data, the best round-trip BLEU is achieved. However, the performance on the validation set is still worse than the baseline.

	Valid (multi-way)		Round-Trip (En-original)
	En⇒Liv	Liv⇒En	
M2M04 (T=5) +CMEA	23.0	28.4	23.4
Add synthetic data and retrain			
En-original	17.2	17.5	30.7
Liv-original	21.5	27.4	25.8
Both	17.0	19.3	32.7

Table 6: Translation performance after adding the synthetic data and retraining the model.

As described in Section 3.2, we consider round-trip BLEU the more appropriate evaluation in this competition due to more data, more general domain, and the same original language as the WMT22 En-Liv test set. Therefore, we used both kinds of synthetic data in our submissions.

4.4 Fine-tuning & Post-processing

Fine-tuning To further exploit the bilingual and monolingual data, we fine-tuned the model on the En⇔Liv validation set for 500 steps jointly with online back-translation on monolingual data.

Post-processing We apply the following rule-based post-processing:

- Apply NFC normalization.
- Replace all the `httpshttp` with `https://`.
- Replace <unk> with empty string.
- When a comma appears between two digits, replace it with a decimal point (only for Liv).
- Regenerate the sentences that detected as repetition with no-repeat constraint[15] (only for Liv).

Final results Table 7 shows the test set performance and round-trip BLEU after fine-tuning and post-processing. As seen, fine-tuning significantly improves model performance on both test set and round-trip BLEU. Post-processing further boosts the performance on the test set.

	Test Set En-Liv		Round-Trip BLEU
	⇒	⇐	
Before fine-tuning	15.8	29.4	32.7
+Fine-tuning	16.3	30.1	37.1
+Post-proc.	17.0	30.4	37.1

Table 7: Translation performance after fine-tuning and post-processing.

5 Conclusion

This paper presents the Tencent AI Lab - Shanghai Jiao Tong University (TAL-SJTU) Low-Resource Translation systems for the WMT22 shared task. We start from the M2M100 1.2B model and investigate techniques to adapt it to English⇔Livonian. We propose cross-model word embedding alignment that transfer the embeddings of Liv4ever-MT to M2M100, enabling it to support Livonian. Then, Estonian and Latvian are involved in model training and synthetic data generation as auxiliary and pivot languages. We further fine-tune the model with validation set and online back-translation followed by rule-based post-processing. In model evaluation, we correct the inaccurate evaluation of Livonian due to inconsistent Unicode normalization and use round-trip BLEU as an alternative to the standard validation set.

Acknowledgements

Zhiwei He and Rui Wang are with MT-Lab, Department of Computer Science and Engineering, School of Electronic Information and Electrical Engineering, and also with the MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200204, China. Rui is supported by General Program of National Natural Science Foundation of China (6217020129), Shanghai Pujiang Program (21PJ1406800), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102). Zhiwei is supported by CCF-Tencent Open Fund (RAGR20210119).

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 Computa-*

[15] We use `--no-repeat-ngram-size 2` in fairseq-generate.

tional Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics.

Ankur Bapna, Isaac Caswell, Julia Kreutzer, Orhan Firat, Daan van Esch, Aditya Siddhant, Mengmeng Niu, Pallavi Baljekar, Xavier Garcia, Wolfgang Macherey, et al. 2022. Building machine translation systems for the next thousand languages. *arXiv preprint arXiv:2205.03983*.

G. Décsy. 1965. *Einführung in die finnisch-ugrische Sprachwissenschaft*. O. Harrassowitz.

Richard Everson. 1998. Orthogonal, but not orthonormal, procrustes problems. *Advances in computational Mathematics*, 3(4).

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. *Journal of Machine Learning Research*.

Xavier Garcia, Noah Constant, Ankur Parikh, and Orhan Firat. 2021. Towards continual learning for multilingual machine translation via vocabulary substitution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.

Zhiwei He, Xing Wang, Rui Wang, Shuming Shi, and Zhaopeng Tu. 2022. Bridging the data gap between training and inference for unsupervised neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael Lyu, and Zhaopeng Tu. 2020. Data rejuvenation: Exploiting inactive training examples for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Wenxiang Jiao, Xing Wang, Shilin He, Zhaopeng Tu, Irwin King, and Michael R Lyu. 2022. Exploiting inactive examples for natural language generation with data rejuvenation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30.

Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael Lyu, and Irwin King. 2021. Self-training sampling with monolingual data uncertainty 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)*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018b. Word translation without parallel data. In *International Conference on Learning Representations*.

Matīss Rikters, Marili Tomingas, Tuuli Tuisk, Valts Ernštreits, and Mark Fishel. 2022. Machine translation for Livonian: Catering to 20 speakers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*.

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)*.

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*. Association for Computational Linguistics.

Longyue Wang, Mu Li, Fangxu Liu, Shuming Shi, Zhaopeng Tu, Xing Wang, Shuangzhi Wu, Jiali Zeng, and Wen Zhang. 2021a. Tencent translation system for the wmt21 news translation task. In *Proceedings of the Sixth Conference on Machine Translation*.

Xing Wang, Zhaopeng Tu, and Shuming Shi. 2021b. Tencent ai lab machine translation systems for the wmt21 biomedical translation task. In *Proceedings of the Sixth Conference on Machine Translation*.

Terry Yue Zhuo, Qiongkai Xu, Xuanli He, and Trevor Cohn. 2022. Rethinking round-trip translation for automatic machine translation evaluation. *arXiv preprint arXiv:2209.07351*.

	XX⇒En			**XX⇒Et**			**XX⇒Lv**			**XX⇒Liv**			**Avg.**
XX	**Et**	**Lv**	**Liv**	**En**	**Lv**	**Liv**	**En**	**Et**	**Liv**	**En**	**Et**	**Lv**	
							All						
Liv4ever-MT (Rikters et al.)	26.17	21.53	19.01	19.48	22.38	23.05	20.85	23.44	25.24	11.03	16.40	17.65	20.52
Our Eval.	25.90	17.94	18.90	19.28	22.31	22.86	20.20	23.31	24.88	10.90	16.62	17.69	20.07
+ Norm Ref.	**26.20**	**18.06**	**19.26**	**20.72**	**24.28**	**24.42**	**24.10**	**27.77**	**29.33**	**14.31**	**20.51**	**22.35**	**22.61**
							Subset (Satversme)						
Liv4ever-MT (Rikters et al.)	-	-	24.49	-	-	-	-	-	-	7.69	-	-	-
Our Eval.	27.50	19.77	24.68	16.69	20.22	18.68	16.05	15.10	19.38	7.58	7.18	9.23	16.83
+ Norm Ref.	**28.45**	**20.21**	**25.76**	**21.41**	**26.74**	**23.75**	**29.10**	**29.82**	**33.56**	**18.23**	**19.87**	**24.15**	**25.09**

Table 8: BLEU scores of Liv4ever-MT on liv4ever-test. **Liv4ever-MT** (Rikters et al.): copied from Rikters et al. (2022). **Our Eval.**: We use the released Liv4ever-MT to generate translation outputs and re-evaluate them with the original references, which shows similar results compared with Rikters et al. (2022). **+ Norm. Ref.**: re-evaluation after normalizing the encoding of references to NFKC.

A Re-evaluating Liv4ever-MT

Table 8 shows the results of re-evaluating Liv4ever-MT on all language pairs. Normalizing references to NFKC improves the average BLEU scores by +2.54 on the entire set and +8.26 on the Satversme subset. It is worth mentioning that liv4ever-test contains data from the following sources: Facebook, Livones.net, Dictionary, Trilium, Stalte, JEFUL and Satversme. However, there does not exist the Unicode inconsistency problem in the other sources except Satversme.