Towards Continual Learning for Multilingual Machine Translation via Vocabulary Substitution

Xavier Garcia and **Noah Constant** and **Ankur P. Parikh** and **Orhan Firat**
Google Research
Mountain View
California
{xgarcia,nconstant,aparikh,orhanf}@google.com

Abstract

We propose a straightforward vocabulary adaptation scheme to extend the language capacity of multilingual machine translation models, paving the way towards efficient continual learning for multilingual machine translation. Our approach is suitable for large-scale datasets, applies to distant languages with unseen scripts, incurs only minor degradation on the translation performance for the original language pairs and provides competitive performance even in the case where we only possess monolingual data for the new languages.

1 Introduction

The longstanding goal of multilingual machine translation (Firat et al., 2016; Johnson et al., 2017; Aharoni et al., 2019; Gu et al., 2018) has been to develop a universal translation model, capable of providing high-quality translations between any pair of languages. Due to limitations on the data available, however, current approaches rely on first selecting a set of languages for which we have data and training an initial translation model on this data jointly for all languages in a multi-task setup. In an ideal setting, one would continually update the model once data for new language pairs arrives. This setting, dubbed in the literature as *continual learning* (Ring, 1994; Rebuffi et al., 2017; Kirkpatrick et al., 2017; Lopez-Paz and Ranzato, 2017), introduces new challenges not found in the traditional multi-task setup, most famously *catastrophic forgetting* (McCloskey and Cohen, 1989), in which the model may lose its previously-learned knowledge as it learns new language pairs. This situation is further complicated by the training procedures of standard tokenizers, such as Byte-Pair Encoding (BPE) (Sennrich et al., 2015b) or Sentencepiece (Kudo and Richardson, 2018), which necessitate access to monolingual data for all the languages considered before producing the vocabulary. Failing to comply with these requirements, one risks suboptimal segmentation rules which in the worst case could result in strings of entirely <UNK> tokens for text in a previously-unseen alphabet.

In this work, we investigate how vocabularies derived from BPE transform if they are rebuilt with the same settings but with additional data from a new language. We show in Section 3.1 that there is a large token overlap between the original and updated vocabularies. This large overlap allows us to retain the performance of a translation model after replacing its vocabulary with the updated vocabulary that additionally supports a new language.

Past works have explored adapting translation models to new languages, typically focusing on related languages which share similar scripts (Gu et al., 2018; Neubig and Hu, 2018; Lakew et al., 2019; Chronopoulou et al., 2020). These works usually focus solely on learning the new language pair, with no consideration for catastrophic forgetting. Moreover, these works only examine the setting where the new language pair comes with parallel data, despite the reality that for a variety of low-resource languages, we may only possess high-quality monolingual data with no access to parallel data. Finally, unlike our approach, these approaches do not recover the vocabulary one would have built if one had access to the data for the new language from the very beginning.

Having alleviated the vocabulary issues, we study whether we are able to learn the new language pair rapidly and accurately, matching the performance of a model which had access to this data at the beginning of training. We propose a simple adaptation scheme that allows our translation model to attain competitive performance with strong bilingual and multilingual baselines in a small amount of additional gradient steps. Moreover, our model retains most of the translation quality on the original language pairs it was trained on, exhibiting no signs of catastrophic forgetting.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1184–1192
June 6–11, 2021.

2 Continual learning via vocabulary substitution

Related works Adapting translation models to new languages has been studied in the past. Neubig and Hu (2018) showed that a large multilingual translation model trained on a subset of languages of the TED dataset (Qi et al., 2018) could perform translation on the remaining (related) languages. Tang et al. (2020) was able to extend the multilingual translation model mBART (Liu et al., 2020) from 25 to 50 languages by exploiting the fact that mBART's vocabulary already supported those additional 25 languages. (Escolano et al., 2021) was able to add new languages to machine translation models by training language-specific encoders and decoders. Other works (Zoph et al., 2016; Lakew et al., 2018, 2019; Escolano et al., 2019) have studied repurposing translation models as initializations for bilingual models for a target low-resource language pair. Most recently (Chronopoulou et al., 2020) examined reusing language models for high-resource languages as initializations for unsupervised translation models for a related low-resource language through the following recipe: build vocabulary $\mathcal{V}_X$ and a language model for high-resource language X; once data for low-resource language Y arrives, build a joint vocabulary $\mathcal{V}_{X,Y}$ and let $\mathcal{V}_{Y|X}$ be the tokens from Y that appear in $\mathcal{V}_{X,Y}$; substitute the vocabulary for the language model with the one given by $\mathcal{V}_X \cup \mathcal{V}_{Y|X}$ and use the language model as the initialization for the translation model.

Our approach In this work, we are not only interested in the performance of our multilingual translation models on new language pairs, we also require that our models *retain* the performance on the multiple language pairs that they were initially trained on. We will also be interested in how the performance of these models compares with those obtained in the oracle setup where we have all the data available from the start. The approaches discussed above generate vocabularies that are likely different (both in selection and number of tokens) from the vocabulary one would obtain if one had *a priori* access to the missing data, due to the special attention given to the new language. This architectural divergence will only grow as we continually add new languages, which inhibits the comparisons to the oracle setup. We eliminate this mismatch by first building a vocabulary $\mathcal{V}_N$ on the N languages available, then once the new language arrives, build a new vocabulary $\mathcal{V}_{N+1}$ as we would have if we had possessed the data from the beginning and replace $\mathcal{V}_N$ with $\mathcal{V}_{N+1}$. We then reuse the embeddings for tokens in the intersection[1] and continue training.

The success of our approach relies on the fact for large N (i.e. the multilingual setting), $\mathcal{V}_N$ and $\mathcal{V}_{N+1}$ are mostly equivalent, which allows the model to retain its performance after we substitute vocabularies. We verify this in the following section.

3 Experiments

In this section, we outline the set of experiments we conducted in this work. We first discuss the languages and data sources we use for our experiments. We then provide the training details for how we trained our initial translation models. Next, we compute the token overlap between various vocabularies derived from BPE before and after we include data for a new language and empirically verify that this overlap is large if the vocabulary already suppots a large amount of languages. We then examine the amount of knowledge retained after vocabulary substitution by measuring the degradation of the translation performance on the original language pairs from replacing the original vocabulary with an updated one. Finally, we examine the speed and quality of the adaptation to new languages under various settings.

Languages considered Our initial model will have to access to data coming from 24 languages[2]. Our monolingual data comes primarily from the *newscrawl* datasets[3] and Wikipedia, while the parallel data comes WMT training sets and Paracrawl. We will adapt our model to the following four languages: **Kazakh**, which is not related linguistically to any of the original 24 languages, but does share scripts with Russian and Bulgarian; **Bengali**, which is related to the other Indo-Aryan languages but possesses a distinct script; **Polish**, which is related to (and shares scripts with) many of the Slavic languages in our original set; **Pashto**, which

[1] Tokens shared between the two vocabularies are also forced to share the same indices. The remaining tokens are rewritten but we still reuse the outdated embeddings.

[2] In alphabetical order: Bulgarian, Czech, Danish, German, Greek, English, Spanish, Estonian, Finnish, French, Gujarati, Hindi, Croatian, Hungarian, Italian, Lithuanian, Latvian, Portugese, Romanian, Russian, Slovak, Slovenian, Tamil.

[3] http://data.statmt.org/news-crawl/

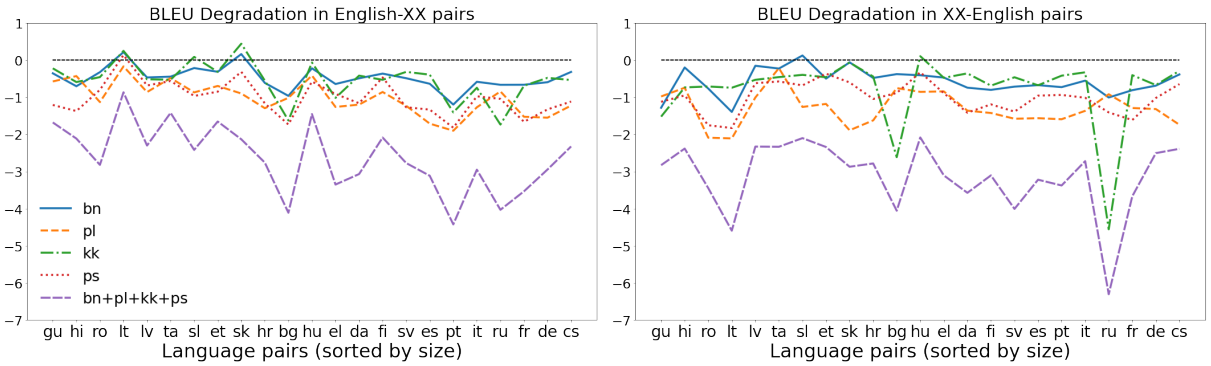

Figure 1: **The degradation in BLEU from substituting vocabularies at inference**. The black dashed line represents the performance from the model trained with the modified vocabulary from the beginning, while the curves represent the BLEU scores from the original model using the new vocabulary at inference.

is not closely related[4] to any of the languages in our original set and has a distinct script. We provide an in-depth account of the data available for each language in the appendix.

Model configurations We perform our experiments in JAX (Bradbury et al., 2018), using the neural network library FLAX[5]. We use Transformers (Vaswani et al., 2017) as the basis of our translation models. We use the Transformer Big configuration and a shared BPE model of 64k tokens with byte-level fallback using the Sentencepiece[6] library. We used a maximum sequence length of 100, discarded all sequences longer than that during training.

Sampling scheme We train our models leveraging both monolingual and parallel datasets, following previous work (Siddhant et al., 2020; Garcia et al., 2020). We sample examples from monolingual and parallel sources with equal probability. Within each source, we use a temperature-based sampling scheme based on the numbers of samples of the relevant datasets with a temperature of 5 (Arivazhagan et al., 2019).

Training objectives We apply the MASS objective (Song et al., 2019) on the monolingual data and cross-entropy on the parallel data. We used the Adam(Kingma and Ba, 2015) optimizer, with an initial learning rate of 4e-4, coupled with a linear warmup followed by a linear decay to 0. The initial warmup took 1k steps, and the total training time was 500k steps. We also included weight decay with a hyperparameter of 0.2.

[4]Closest languages are in the Indic branch, but the Indic and Iranian branches split over 4000 years ago.

[5]https://github.com/google/flax

[6]We use 1.0 character coverage, split by whitespace, digits, and include a special token `MASK` for the MASS objective.

# langs in base	*bn*	*pl*	*kk*	*ps*	*bn+pl +kk+ps*
1	53.5%	47.0%	46.0%	47.8%	24.4%
5	84.0%	80.8%	81.8%	80.2%	57.7%
10	90.3%	87.4%	89.3%	87.2%	70.9%
15	93.1%	91.8%	90.7%	90.5%	76.9%
20	94.8%	90.1%	93.0%	93.1%	79.2%
24	95.4%	94.3%	95.2%	93.5%	82.7%

Table 1: **Percentage of token overlap between vocabularies before & after the inclusion of a new language.** We denote the case where we add all the unseen languages by the column 'bn+pl+kk+ps'.

Evaluation We use beam search with a beam size of 4 and a length penalty of $\alpha = 0.6$ for decoding. We evaluate the quality of our models using BLEU scores (Papineni et al., 2002). We exclusively use detokenized BLEU, computed through sacreBLEU (Post, 2018) for consistency with previous work and future reproducibility.[7]

3.1 Transfer learning from vocabulary substitution

Measuring token overlap We now examine the impact on the vocabulary derived from a BPE model upon the inclusion on a new language. We first build corpora consisting of text[8] from 1, 5, 10, 15, 20, and 24 of our original languages. For each corpus, we make copies and add additional data for either Bengali, Polish, Kazakh, Pashto, or their union, yielding a total of 30 corpora. We build BPE models using the same settings for each corpus, compute the token overlap between the vocabularies with and without the additional language, and

[7]BLEU + case.mixed + numrefs.1 + smooth.exp + tok.13a + version.1.4.14

[8]We used 1 million lines of raw text per language.

	Model	*PMIndia* bn↔en		*newsdev2020* pl↔en		*newstest2019* kk↔en		*FLoRes devset* ps↔en	
Original Vocabulary	Unadapted	0.0	0.0	2.4	4.0	0.7	2.2	0.0	0.0
	xx monolingual & parallel	5.7	13.6	20.2	26.2	3.9	17.2	2.8	10.3
	4xx monolingual & parallel	5.3	15.1	18.3	25.0	2.7	15.8	2.3	8.4
Adapted Vocabulary	xx monolingual	0.0	1.7	13.9	24.3	0.9	19.0	0.0	6.5
	xx monolingual (+BT)	-	-	21.3	24.1	4.7	19.5	-	-
	xx monolingual & parallel	10.0	27.2	**21.5**	**27.5**	**5.9**	20.2	6.6	15.1
	4xx monolingual & parallel	**10.5**	26.4	20.3	26.8	5.6	**20.5**	**6.7**	**15.2**
Oracle	xx monolingual & parallel	10.1	**29.2**	19.6	26.8	5.4	**20.5**	6.6	14.7
	4xx monolingual & parallel	10.0	28.6	18.9	26.4	5.4	20.3	6.2	14.4

Table 2: **BLEU scores on the new languages.** The "monolingual" models have access to exclusively monolingual data for the new language(s), while "monolingual & parallel" models add parallel data as well. Models with "xx" add a single language, while "4xx" models add four languages together.

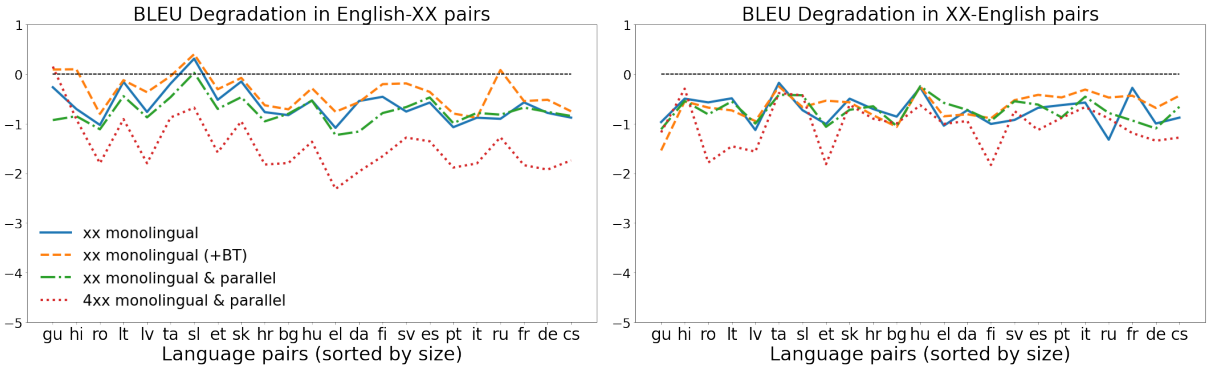

Figure 2: **Measuring forgetting after adaptation.** The difference in BLEU for the original language pairs between the oracle model and models adapted to Kazakh.

report the results in Table 1. In the multilingual setting, we attain large token overlap, more than 90%, even for languages with distinct scripts or when we add multiple languages at once. We extend this analysis to different vocabulary sizes and examine which tokens are "lost" in Appendix A.3.

3.2 Evaluating translation quality and catastrophic forgetting

Measuring the deterioration from swapping vocabularies at inference To measure the amount of knowledge transferred through the vocabulary substitution, we compute the translation performance of our initial translation model with the adapted vocabularies *without any additional updates*. For each new language, we compute the change in BLEU from the model with its original vocabulary and the one utilizing the adapted one and plot the results in Figure 1. Notably, we only incur minor degradation in performance from the vocabulary substitution.

We now study the effect of introducing a new language into our translation model. We require an adaptation recipe which enjoys the following properties: *fast*, in terms of number of additional gradient steps; *performant*, in terms of BLEU scores on the new language pair; *retentive*, in terms of minimal regression in the translation performance of the model on the original language pairs.

Our solution: re-compute the probabilities for the temperature-based sampling scheme using the new data, upscale the probabilities of sampling new datasets by a factor then rescale the remaining probabilities so that their combined sum is one. We limit ourselves to either 15k or 30k additional steps (3% and 6% respectively of the training time for the original model) depending on the data available[9] to ensure fast adaptation. We reset the Adam optimizer's stored accumulators, reset the learning rate to 5e-5 and keep it fixed. We provide more details in Appendix A.2. Aside from these modifications, we continue training with the same objectives as before unless noted otherwise. We include the results for oracle models trained in the same way as the original model but with access to both the adapted

[9]We use 15k steps if we leverage both monolingual and parallel data for a single language pair. We use 30k steps if we only use monolingual data or if we are adapting to all four languages at once.

vocabulary *and* the missing data. We compute the BLEU scores and report them in Table 2.

Our models adapted with parallel data are competitive with the oracle models, even when we add all four languages at once and despite the restrictions we imposed on our adaption scheme. For languages that share scripts with the original ones (Kazakh and Polish), we can also attain strong performance leveraging monolingual data alone, albeit we need to introduce back-translation (Sennrich et al., 2015a) for optimal performance. We can also adapt the translation model using the original vocabulary, but the quality lags behind the models using the adapted vocabularies. This gap is larger for Bengali and Pashto, where the model is forced to rely on byte-level fallback, further reaffirming the value of using the adapted vocabularies.

To examine whether catastrophic forgetting has occured, we proceed as in Section 3.1 and examine the performance on the original language pairs after adaptation on the new data against the oracle model which had access to this data in the beginning of training. We present the results for the models adapted to Kazakh in Figure 2. All the models' performance on the original language pairs deviate only slightly from the oracle model, mitigating some of the degradation from the vocabulary substitution i.e. compare the kk and bn+pl+kk+ps curves in Figure 1 to the curves in Figure 2.

Lastly, we compare our models with external baselines for Kazakh. We consider the multilingual model mBART (Liu et al., 2020) as well as all the WMT submissions that reported results on English ↔ Kazakh. Of these baselines, only mBART and (Kocmi et al., 2018) use sacreBLEU which inhibits proper comparison with the rest of the models. We include them for completeness. We report the scores in Table 3. Our adapted models are able to outperform mBART in both directions, and as well some of the weaker WMT submissions, despite those models specifically optimizing for that language pair and task.

4 Conclusion

We present an approach for adding new languages to multilingual translation models. Our approach allows for rapid adaptation to new languages with distinct scripts with only a minor degradation in performance on the original language pairs.

	Model	*newstest2019* kk↔en	
Without $en \leftrightarrow kk$	*xx monolingual*	0.9	19.0
	xx monolingual (+BT)	4.7	19.5
With $en \leftrightarrow kk$	Kocmi and Bojar (2019)	8.7	18.5
	Li et al. (2019)	11.1	**30.5**
	Casas et al. (2019)	**15.5**	21.0
	Dabre et al. (2019)	6.4	26.4
	Briakou and Carpuat (2019)	-	9.94
	Littell et al. (2019)	-	25.0
	mBART (Liu et al., 2020)	2.5	7.4
	xx monolingual & parallel	5.9	20.2
	4xx monolingual & parallel	5.6	20.5

Table 3: **BLEU scores on the new languages against external baselines.** The models in italics are ours.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *NAACL*.

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

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. 2018. Jax: composable transformations of python+numpy programs.

Eleftheria Briakou and Marine Carpuat. 2019. The university of maryland's kazakh-english neural machine translation system at wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 134–140.

Noe Casas, José AR Fonollosa, Carlos Escolano, Christine Basta, and Marta R Costa-jussà. 2019. The talp-upc machine translation systems for wmt19 news translation task: pivoting techniques for low resource mt. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 155–162.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2020. Reusing a pretrained language model on languages with limited corpora for unsupervised nmt. *arXiv preprint arXiv:2009.07610*.

Raj Dabre, Kehai Chen, Benjamin Marie, Rui Wang, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. 2019. Nict's supervised neural machine translation systems for the wmt19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 168–174.

Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. 2019. From bilingual to multilingual neural machine translation by incremental training. In

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 236–242, Florence, Italy. Association for Computational Linguistics.

Carlos Escolano, Marta R Costa-Jussà, and José AR 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.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL*.

Xavier Garcia, Pierre Foret, Thibault Sellam, and Ankur P Parikh. 2020. A multilingual view of unsupervised machine translation. *arXiv preprint arXiv:2002.02955*.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor OK Li. 2018. Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*.

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

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

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Tom Kocmi and Ondřej Bojar. 2019. CUNI submission for low-resource languages in WMT news 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 234–240, Florence, Italy. Association for Computational Linguistics.

Tom Kocmi, Roman Sudarikov, and Ondřej Bojar. 2018. CUNI submissions in WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 431–437, Belgium, Brussels. Association for Computational Linguistics.

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

Surafel M Lakew, Aliia Erofeeva, Matteo Negri, Marcello Federico, and Marco Turchi. 2018. Transfer learning in multilingual neural machine translation with dynamic vocabulary. *arXiv preprint arXiv:1811.01137*.

Surafel M Lakew, Alina Karakanta, Marcello Federico, Matteo Negri, and Marco Turchi. 2019. Adapting multilingual neural machine translation to unseen languages. *arXiv preprint arXiv:1910.13998*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, et al. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266.

Patrick Littell, Chi-kiu Lo, Samuel Larkin, and Darlene Stewart. 2019. Multi-source transformer for kazakh-russian-english neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 267–274.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*, pages 6467–6476.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. *arXiv preprint arXiv:1808.04189*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ye Qi, Devendra Singh Sachan, Matthieu Felix, Sarguna Janani Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? *arXiv preprint arXiv:1804.06323*.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Mark Bishop Ring. 1994. *Continual learning in reinforcement environments*. Ph.D. thesis, University of Texas at Austin Austin, Texas 78712.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2827–2835, Online. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575.

A Appendix

A.1 Data statistics and details

We outline the counts, domains, test set, and BLEU scores of our original translation model on the 24 languages in Table 6. We do the same for the unseen languages in Table 7. All the Paracrawl data is from v6.0.

A.2 Adaption schemes

We now explain in detail our configurations:

Monolingual data for a single language In this case, we compute the probabilities following the temperature-based sampling scheme that we would have obtained had we computed with this data in the first place. Then we proceed to set the sampling probability of the new monolingual to 30% and rescale the remaining probabilities so that they add up to 1.

Monolingual data for a single language coupled with back-translation In order to properly utilize back-translation, we first train the model for 10k step in the same fashion as the previous paragraph. Then, we use offline backtranslation on the new monolingual data to generate pseudo-parallel data. We then treat this data as authentic and include it in the model. We set the sampling probability of the pseudo-parallel data to be 10%, we reset the sampling probability of the monolingual data to 10%, and rescale the rest so that they sum up to 1. We then continue training for an additional 20k steps, amounting to a total of 30k steps.

Monolingual & parallel data for a single language We multiply the probabilities of the new parallel data by a factor of 10, set the sampling probability of the monolingual data to 10% then rescale the reamining probabilities so that they are normalized. We then train for 15k steps.

Monolingual & parallel data for all four languages We do not use the same scaling as before, since this would aggressively undersample the original language pairs. Instead, we first average the total probabilities for the new parallel data, multiply it by 5 and then assign this probability to each of the parallel datasets. We then fix the probability of sampling the new monolingual datasets to be 5% each. We then train for 30k steps

A.3 Token overlap analysis

We first verify that the results in Table 1 apply for different vocabulary sizes. We compute analogous tables for vocabulary size of 32k and 128k tokens in Table 4 and 5 respectively.

Next, we examine which tokens are lost during the vocabulary substitution. Since the Sentencepiece library does not provide an easy way to acquire frequency scores for BPE models after training, we instead use the order of the tokens as a proxy for the relative ranking obtained by sorting the tokens by frequency. For each language, we produce violin plots for the indices in the original

# languages in base model	*bn*	*pl*	*kk*	*ps*	*bn+pl+kk+ps*
1	53.3%	49.3%	48.4%	47.7%	22.8%
5	83.0%	81.1%	81.6%	78.0%	51.9%
10	89.7%	87.4%	88.9%	85.6%	65.1%
15	92.8%	92.1%	90.2%	88.9%	72.1%
20	94.7%	90.3%	92.9%	92.2%	79.0%
24	95.6%	95.3%	95.5%	93.2%	83.8%

Table 4: **Token overlap between vocabularies (consisting of 32k tokens) before & after the inclusion of a new language.**

# languages in base model	*bn*	*pl*	*kk*	*ps*	*bn+pl+kk+ps*
1	53.5%	45.1%	43.5%	47.2%	21.1%
5	85.1%	80.7%	81.6%	81.7%	54.0%
10	91.0%	87.5%	89.3%	88.4%	67.4%
15	93.6%	91.8%	91.4%	91.5%	74.6%
20	95.1%	90.3%	93.3%	93.5%	79.4%
24	95.5%	94.2%	95.4%	93.9%	82.8%

Table 5: **Token overlap between vocabularies (consisting of 128k tokens) before & after the inclusion of a new language.**

vocabulary which are not in the adapted vocabulary for that language in Figure 3.

Critically, we observe that most of the tokens lost are towards the end of spectrum, suggesting that the model is mostly discarding infrequent tokens. Notably, it cannot discard the tail due to our requirement of full character coverage, which introduces a variety of rare Unicode characters as tokens that reside in the tail.

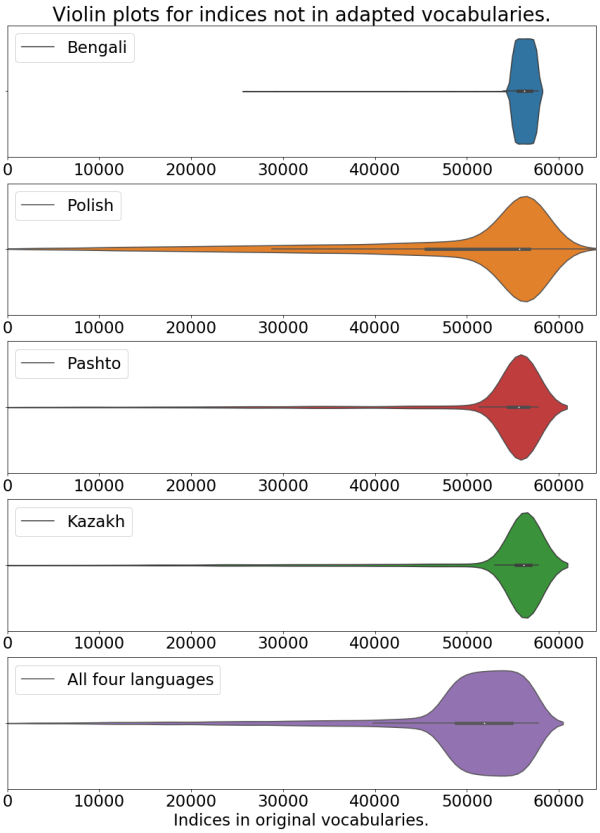

Figure 3: **Violin plots for the indices in the original vocabulary that do not appear in the adapted vocabulary** Note that all language configurations, most of the indices that do not appear in the overlap are towards the infrequent side of the spectrum.

Language	Monolingual data (# of lines)	Parallel data (# of examples)	Domain (Monolingual data)	Domain (Parallel data)	Test set	Language family	BLEU en-xx	BLEU xx-en
Bg	39610418	4111172	NewsCrawl	Paracrawl	TED	Slavic	32.43	35.77
Cs	81708712	64336053	NewsCrawl	WMT	WMT'18	Slavic	18.42	28.60
Da	4139992	6370432	Wiki	Paracrawl	TED	Germanic	38.81	42.87
De	333313278	4508785	NewsCrawl	WMT	WMT'14	Germanic	23.63	30.38
El	8332782	5298946	NewsCrawl	Paracrawl	TED	Hellenic	29.03	34.40
Es	53874815	15182374	NewsCrawl	WMT	WMT'13	Romance	31.74	33.23
Et	5367030	2175873	NewsCrawl	WMT	WMT'18	Uralic	16.99	27.53
Fi	21520558	6587448	NewsCrawl	WMT	WMT'19	Uralic	16.95	27.08
Fr	87063385	40853298	NewsCrawl	WMT	WMT'14	Romance	35.04	36.13
Gu	816575	155798	NewsCrawl	WMT	WMT'19	Indo-Aryan	10.92	20.91
Hi	23611899	313748	NewsCrawl	WMT	WMT'14	Indo-Aryan	13.36	18.98
Hr	6814690	6814690	NewsCrawl	Paracrawl	TED	Slavic	25.31	34.81
Hu	40879784	4963481	NewsCrawl	Paracrawl	TED	Uralic	15.90	24.25
It	2836989	2747344	NewsCrawl	Paracrawl	TED	Romance	31.87	36.59
Lt	2836989	635146	NewsCrawl	WMT	WMT'19	Baltic	11.56	30.82
Lv	11338472	637599	NewsCrawl	WMT	WMT'17	Baltic	17.16	22.69
Pt	9392574	20677300	NewsCrawl	Paracrawl	TED	Romance	33.25	41.79
Ro	21033306	610320	NewsCrawl	WMT	WMT'16	Romance	27.18	36.92
Ru	93827187	38492126	NewsCrawl	WMT	WMT'19	Slavic	22.20	34.70
Sk	3040748	3303841	Wiki	Paracrawl	TED	Slavic	22.59	29.52
Sl	2669157	1923589	Wiki	Paracrawl	TED	Slavic	21.06	25.73
Ta	708500	736479	NewsCrawl	WMT	WMT'20	Dravidian	6.29	12.06

Table 6: Details on the original 24 languages considered. For Tamil, we did not have access to the test set, so we used newsdev2019 instead. The BLEU scores are from the our original translation model.

Language	Monolingual data (# of lines)	Parallel data (# of examples)	Domain (Monolingual data)	Domain (Parallel data)	Test set	Language family
bn	3918906	27584	Newscrawl	PMIndia	PMIndia	Indo-Aryan
kk	4032908	222424	Newscrawl + Wiki Dumps	WMT	WMT	Turkic
pl	3788276	5001447	Newscrawl	WMT	WMT	Slavic
ps	6969911	1134604	Newscrawl + CommonCrawl	WMT	WMT	Indo-Iranian

Table 7: Details on the additional 4 languages considered for adaptation. For Polish, we did not have access to the test set so we used the dev set instead.