Low-resource Machine Translation

Deep Natural Language Processing

Andrea Cavallo
Politecnico di Torino
Torino, Italy
s287965@studenti.polito.it

Lorenzo Scarciglia
Politecnico di Torino
Torino, Italy
lorenzo.scarciglia@studenti.polito.it

Cristina Tortia
Politecnico di Torino
Torino, Italy
cristina.tortia@studenti.polito.it

***Abstract*—This work focuses on the problem of low-resource machine translation, i.e. machine translation with languages for which a large corpus of parallel text is not available. Since sequence-to-sequence models require a lot of samples to be trained from scratch, this is not feasible for these languages, and an alternative approach must be applied. In this paper, we investigate the impact of multilingual finetuning of a model pretrained on a large corpus for a different language pair. In particular, we start from a model pretrained on an English-Chinese large parallel corpus, and we finetune it on different low-resource target languages using ALT dataset. The finetuning process involves two steps. In the first phase, called mixed finetuning, the model is finetuned using a dataset containing both low-resource language and Chinese samples. In the second phase, called pure finetuning, the model is finetuned with a dataset containing only samples from the low-resource language. We show that this technique achieves better results with respect to purely finetuning the initial model on the target language, and the final results are better or comparable to the other low-resource machine translation approaches. Furthermore, we extend this approach by performing translation between two low-resource languages and by applying it to a different dataset. Both experiments confirm the effectiveness of the technique. The code for this project is available at the following link: https://github.com/andrea-cavallo-98/Low-resource-Machine-Translation.**

***Index Terms*—Machine Translation, Low-resource languages**

I. Problem statement

The task of machine translation relies mostly on encoder-decoder architectures, that can be trained end-to-end on a bilingual corpus. This approach provides very good performances if a large bilingual corpus is available for the language pair, but cannot achieve comparable performances for low-resource language pairs, i.e. those for which just a small parallel corpus is available. As explained in [1], there are different approaches to tackle this issue.

One option is to have multilingual models (e.g. in [3]), i.e. models that should theoretically be able to translate between any language pair, with parameters that are shared across languages. This helps developing a universal representation space, which is proved to be useful also for low-resource languages.

Another successful approach has been the large-scale pre-training of a multilingual model, such as mBART [4], which demonstrated improvements to machine translation in supervised, unsupervised and semi-supervised (i.e. with back-translation) conditions, including low-resource language pairs.

In conclusion, another possibility is to exploit transfer learning [2], by training a parent model on one language pair, and then using the trained parameters to initialise a child model, which is then trained on the desired low-resource language pair. In this context, Dabre, Fujita, and Chu [5] propose a multi-step approach to target low-resource languages. In particular, an encoder-decoder model is pretrained on a large corpus. Then, it is finetuned on a dataset that contains both sentences from the initial corpus and sentences from the low-resource language pair (*mixed finetuning*). In conclusion, the model is further finetuned on a dataset containing only sentences from the low-resource language pair. Following this approach, the authors achieve significant improvements on 7 low-resource languages, outperforming the previous state of the art.

This work reproduces and extends the experiments[1] of Dabre, Fujita, and Chu [5]. In particular, the two-step finetuning technique is experimented for 4 low-resource languages (Vietnamese, Indonesian, Filipino and Khmer), chosen as those that provided the best results in the original paper. Then, the same approach is applied to finetune a model on the opposite translation task: from a low-resource language (Vietnamese) to English. With this model, the English sentences are then translated into another low-resource language (Indonesian) using the already finetuned model. In conclusion, the approach is experimented also on a different dataset and language pair, to evaluate its generalization capabilities.

[1] configurations #2 and #4

II. Methodology

A. Implementation of multilingual finetuning

In the first part of the project, a translation model pretrained on a large parallel corpus is finetuned on a low-resource language pair. The target languages are selected among those in the ALT dataset, which contains about 20k samples for different Asian languages. Since the exact model used in the paper by Dabre, Fujita and Chu was not available, the starting model was selected to be MarianMT[2], a transformer-based model pretrained on the OPUS dataset on the English-Chinese language pair. Figure 1 reports the sketch of the standard transformer-based encoder-decoder architecture for Machine Translation, which is used by MarianMT.

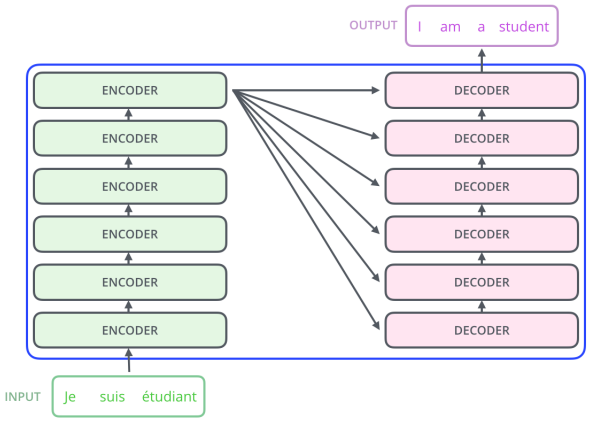


Fig. 1: Sketch of an encoder-decoder architecture

Since the model is available on the Huggingface Transformers library, we exploited the finetuning APIs[3] of the library to perform the experiments. Among those available, we selected 4 target languages, taking into consideration the best results achieved in the paper: Vietnamese, Indonesian, Filipino and Khmer. In particular, we took as reference Dabre et al. experiments #2 and #4[4]. In configuration #2 they use a pre-trained model on the resource-rich language (Chinese) and apply pure finetuning on a target language, while in #4 they perform mixed finetuning before and pure finetuning afterwards. During the mixed finetuning phase, English-Chinese sentences are mixed to sentences from one target language (not all 7 together due to memory and time issues). In the pure finetuning phase, instead, only low-resource languages are used. A sketch of the approach used to generate the dataset for the mixed and the pure finetuning is reported in Figure 2. To allow the model to understand the target language of the current English input sentence, we placed the special token $<2zz>$ in front of the sentence (where *zz* is the ISO code of the target language)[5].
Since the MarianMT model can only accept fixed-length input sentences, we selected the size of the input trying to minimize the number of cut-off sentences, while also considering the training time and the memory issues occurred during training. The selected values for the different languages and the corresponding percentages of cut-off sentences are reported in Table I[6].

Languages	Input length	Cut-off sentences (%)	
		English	Target language
Vietnamese	64	3.4	8.1
Indonesian	128	0.1	0.1
Filipino	128	0.1	0.3
Khmer	128	0.1	0.5

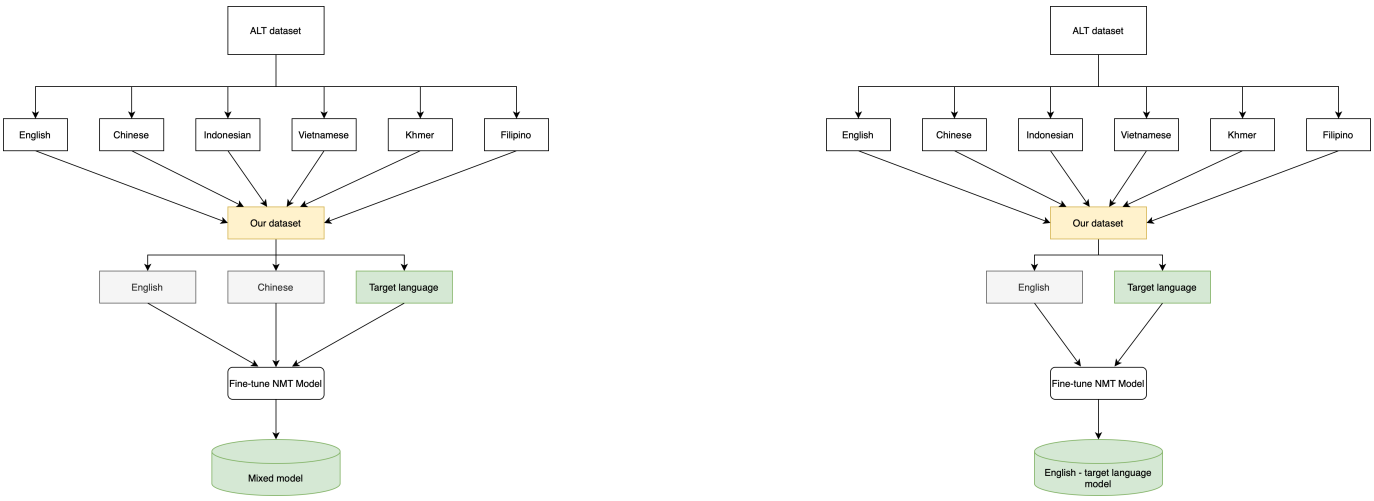
TABLE I: Input length and cut-off sentences for the different languages

Another issue is related to the tokenizer: since the starting model is trained on English and Chinese sentences, the related tokenizer knows nothing about the selected low-resource target languages. Therefore, the sentences in the dataset were tokenized using a different tokenizer (mBART[7]), pretrained on the specific languages. Then, the extracted tokens were added to the initial Marian tokenizer, where they got automatically assigned to an ID and to a random embedding in the model. However, adding new tokens to the Marian tokenizer caused an additional problem: the tokenizer was not able to properly tokenize English sentences anymore. Therefore, to preprocess the dataset, we used a newly instantiated Marian tokenizer to tokenize input sentences (in English), and the extended Marian tokenizer to tokenize the output sentences (in the low-resource target language).
The performances of the model were evaluated using the BLEU score, which is a common choice for machine translation tasks.

B. Extension I: changing direction of the translations

As a first extension, we experimented the impact of this approach on a slightly different task: translating from a low-resource language to English, and then from English to a different low-resource language. As a matter of fact, training a model directly on a parallel corpus for the two low-resource languages is low-performing, since few examples are available for the training. On the other hand, an intermediate translation in English can achieve satisfactory performances.
In particular, starting from MarianMT pretrained on the Chinese-English translation, we finetuned it to perform Vietnamese-English translation. Then, the English sentences obtained with this model were translated to other low-resource languages, using the models finetuned in the first phase of the project. The results of the translations are evaluated using the BLEU score.

[2]https://huggingface.co/Helsinki-NLP/opus-mt-en-zh
[3]https://huggingface.co/docs/transformers/training
[4]Table 1 in [5]
[5]*zh* (Chinese), *tl* (Filipino), *id* (Indonesian), and *km* (Khmer)
[6]Although the percentages of cut-off sentences for Vietnamese are higher, we experimented also some epochs of finetuning using an input size of 128, and the results were very similar. Therefore, we performed all the experiments using an input size of 64, since the training time was significantly lower.
[7]https://huggingface.co/facebook/mbart-large-50



(a) Mixed step

(b) Pure step

Fig. 2: Sketch of the datasets for the mixed and pure finetuning process

C. Extension II: training on a different dataset

As a second extension, we experimented the described approach on a different low-resource language pair using a different dataset. In particular, we chose WikiMatrix [6], a dataset containing parallel sentences for 1620 language pairs. The examples are mined from Wikipedia using a distance metric to align sentences from different languages. This approach allows to gather samples for many language pairs, but provides, in some cases, wrong alignments. This results in a less reliable dataset with respect to ALT, with consequent worse performances in model training. To make results comparable with the previous sections, we chose a language pair for which WikiMatrix provides about 20k sentences: English-Kazakh. As starting models, we experimented two different possibilities: since Kazakh belongs to the Turkic language family, one option is MarianMT pretrained on an English-Turkish large corpus[8]. The other one is MarianMT pretrained on an English-Russian large corpus, as the alphabet used by Kazakh is the same as Russian (Cyrillic)[9].
This translation task was investigated in 2019 in [7] on a different dataset. The authors showed that using data from an additional language, which can be related or not (Turkish and Russian, respectively), improves the performance notably instead of end-to-end training on Kazakh alone.

III. Experiments

A. Datasets

1) ALT parallel corpus: The main task and the first extension rely on the ALT parallel corpus dataset. This dataset is based on 20,000 sentences from English Wikinews that were translated into Asian languages. The dataset consists of 13 languages: English, Bengali, Filipino, Hindi, Bahasa Indonesia (Indonesian), Japanese, Khmer, Lao, Malay, Myanmar (Burmese), Thai, Vietnamese, Chinese (Simplified Chinese). Among the available languages, we used Indonesian, Vietnamese, Filipino (Latin alphabet), and Khmer (abugida[10] script).

2) WikiMatrix: WikiMatrix [6] is the dataset used in the second extension. It is based on Wikipedia articles, has 85 different languages (also including low-resource ones) and 1620 language pairs.
In particular, the dataset is automatically generated by pairing translations of the same Wikipedia pages. For each article, sentences are paired according to a distance measure (cosine distance) using multilingual sentence embeddings. The margin criterion is used to select and extract possible mutual translations. Although this approach allows to automatically generate a huge amount of training data for many language pairs, it should be noticed that not all generated examples are meaningful or correct, and, therefore, the expected performances on this dataset are lower with respect to ALT.

B. Framework

We used Google Colaboratory as training environment, therefore, the execution times depended on the hardware provided. On average, finetuning on a mixed dataset took 4 hours (5 epochs), whereas finetuning on a pure dataset took 2 hours (5 epochs).
We used `Python 3.7.12` and the following libraries:

- `transformers 4.16.2`
- `datasets 1.18.3`
- `metrics 0.3.3`
- `sacrebleu 2.0.0`
- `torch 1.10.0 + cu111`
- `sentencepiece 0.1.96`

[8]https://huggingface.co/Helsinki-NLP/opus-tatoeba-en-tr
[9]https://huggingface.co/Helsinki-NLP/opus-mt-en-ru

[10]https://en.wikipedia.org/wiki/Abugida

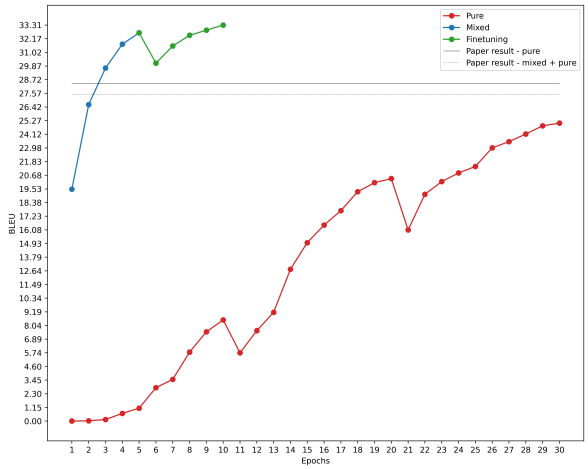


Fig. 3: BLEU score for English-Khmer model, 30 epochs

Each dataset was randomly splitted in `train`, `eval` and `test` set using the `train_test_split` function provided by Scikit-Learn (`seed=42`). Each model was trained for a total of `10 epochs` (except for the pure fine-tuning of the Khmer language, see section III-C) with the following set of parameters:

- `learning rate = 2e-4`
- `batch = 16`
- `weight decay = 0.01`

The evaluation metric used is the BLEU score, which is the precision of *n*-grams of the MT output compared to the reference, weighted by a brevity penalty to penalize overly short translations.

C. Multilingual finetuning

In the first experiment, the pretrained MarianMT model is finetuned on 4 different low-resource target languages in three ways:

- *Pure*: the MarianMT model is finetuned directly on an English-target dataset for 10 epochs
- *Mixed*: the MarianMT model is finetuned for 5 epochs on a dataset containing a mix of English-Chinese and English-target sentences
- *Mixed* + *Pure*: the MarianMT model already finetuned on the mixed dataset is further finetuned on a dataset containing only English-target language sentences

We trained all models using the same learning rate and batch size. The evolution of the BLEU score for the models is reported in Figures 5. It can be noticed that, in general, the mixed finetuning provides better performances sooner with respect to the pure finetuning.

Moreover, applying a pure finetuning to the model trained on the mixed dataset provides a final result which outperforms

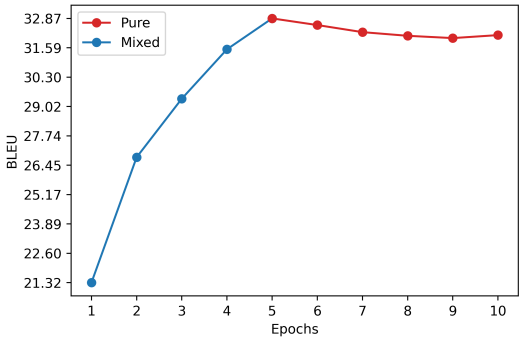


Fig. 4: BLEU score for Vietnamese-English model

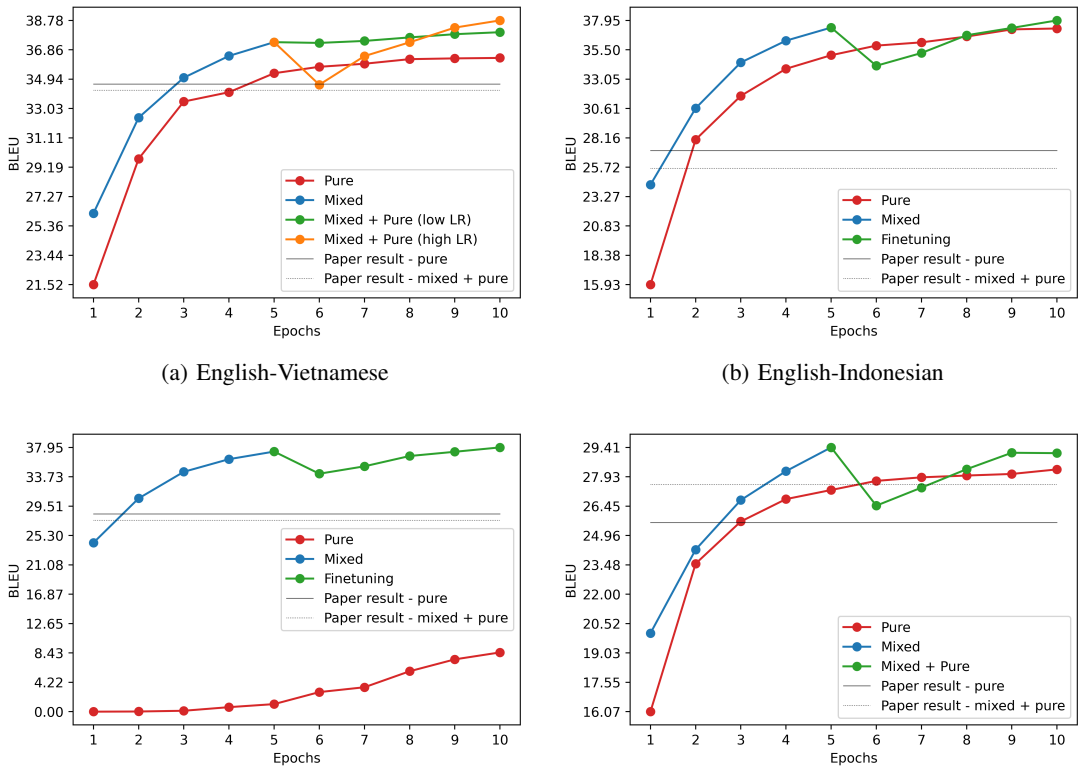
the standard pure finetuning.

It should be noticed that, when starting the pure finetuning on the model finetuned on the mixed dataset, the first epochs achieve a lower BLEU score with respect to the previous results. To investigate this behavior, we tried to perform the pure finetuning on Vietnamese using two different learning rates: `2e-4` and `2e-5`. As the results show, using a low learning rate solves the problem of the initial drop in performances when starting the finetuning. However, since the learning is slower, the final result is worse with respect to the higher learning rate. Therefore, in the other experiments we used the higher learning rate (`2e-4`) also for pure finetuning. With respect to the other languages, pure finetuning on Khmer provided very poor results. Therefore, we tried to train for more epochs, and after 30 epochs we managed to obtain reasonable results. However, this case in particular shows the positive effect of the mixed finetuning approach, since it allows to achieve better results much faster. The evolution of the BLEU score during the complete 30 epochs is reported in Figure 3.

Languages	**BLEU scores**			
	Pure	Mixed	Mixed + Pure	Paper
Vietnamese	36.56	38.07	38.9	34.22
Indonesian	37.27	37.35	37.74	25.62
Filipino	28.3	29.4	29.12	26.61
Khmer	25.06	32.65	33.32	27.49

TABLE II: BLEU score for different finetuned models

Table II reports the final BLEU score on the test dataset for different models. It can be observed that the mixed finetuning always outperforms the pure finetuning, and sometimes mixed+pure finetuning further improves the results. By comparing these results with the corresponding ones reported by Dabre, Fujita, and Chu [5], it is possible to notice that the performances are increased in all cases. However, the experimental setup is not exactly the same, since the pretrained models differ.



(a) English-Vietnamese

(b) English-Indonesian

(c) English-Khmer

(d) English-Filipino

Fig. 5: BLEU score during training for different English-target language pairs

D. Extension I

For the first extension, we extended this approach to create a model that can translate between two low-resource languages using English as bridge. In particular, we fine-tuned the MarianMT Chinese-English model[11] to target the Vietnamese-English language pair. Then, we used the previously finetuned models to translate into another target low-resource language. More specifically, the tested language pairs are Vietnamese-Indonesian and Vietnamese-Khmer.

Figure 4 shows the evolution of the BLEU score when training the model from Vietnamese to English. It can be observed that performing the pure finetuning does not provide any benefit. Therefore, the model used in the following is the one trained on 5 epochs in the mixed setting.

Source language	Target language	BLEU score
Vietnamese	English	33.35
Vietnamese	Indonesian	23.00
Vietnamese	Khmer	24.13

TABLE III: BLEU score for translations among low-resource language pairs

Table III shows BLEU scores for the different language pairs. Results are indeed lower than the ones obtained in section III-C, due to the additional step of English translation, but still satisfactory.

[11]https://huggingface.co/Helsinki-NLP/opus-mt-zh-en

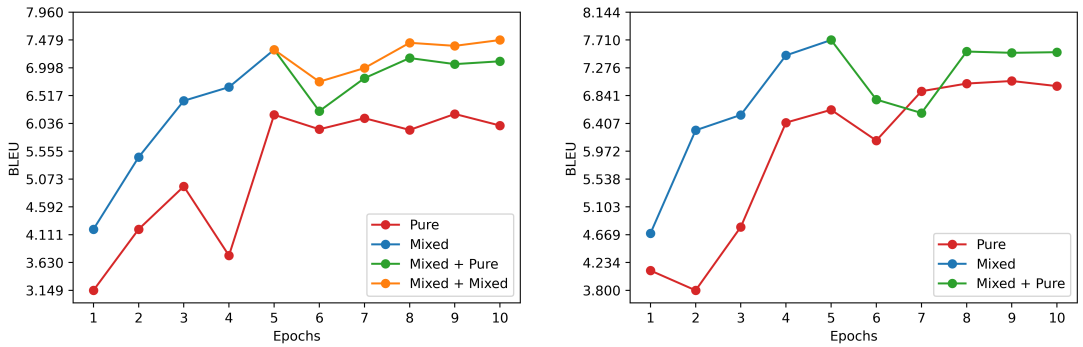
E. Extension II

For the second extension, we tested the proposed approach on a different language pair and on a different dataset. In particular, we selected WikiMatrix and we opted for a language with approximately the same number of examples as ALT: Kazakh. As starting models, we tested two models pretrained on two different high-resource language-pairs (English-Turkish and English-Russian). We chose Turkish because Kazakh and Turkish are both agglutinative languages and belong to the same language family (Turkic). The choice of Russian is, on the other hand, due to the alphabet. Indeed, they use both the Cyrillic alphabet.

Initial model	BLEU scores		
	Pure	Mixed	Mixed + Pure
En-Tk	6.93	7.33	7.26
En-Ru	6.18	7.31	7.11

TABLE IV: BLEU score for En-Kk with different initial models

Table IV reports the results for the trained models. Whilst the pure finetuning shows that the model pretrained on Turkish



(a) English-Russian (b) English-Turkish

Fig. 6: BLEU score during training for English-Kazakh with different pre-trained models

achieves better performances with respect to the one pretrained on Russian, the mixed finetuning provides similar results for both starting models. In both cases, mixed training achieves better performances with respect to pure finetuning, which confirms its effectiveness also in a different context.
To further experiment on Kazakh, we tried to run the mixed finetuning for 10 epochs in total, using the model pretrained on Russian. As shown in Figure 6, the final results are slightly better with respect to the mixed+pure finetuning (7.11[12] vs 7.48[13]). This corroborates the fact that low-resource languages can benefit from mixed trainings.

IV. Conclusions

In this paper we implemented the approach presented by Dabre et al. [5] for low-resource neural machine translation using the MarianMT architecture. We started with a pretrained model which can translate from English to Chinese and we finetuned it on 4 different low-resource target languages using the ALT parallel corpus dataset. For each of the chosen languages, we made two kinds of trainings. One pure fine-tuning (10 epochs) and one mixed[14] (5 epochs) + pure fine-tuning (5 epochs). As expected, the mixed training helps a lot the model to learn how to translate into low-resource language, showing better performances with respect to the plain pure finetuning. In some cases, performing pure finetuning on the mixed finetuned models does not provide any improvement, but in all the reported cases, the mixed finetuning provides better results with respect to pure finetuning.
Moreover, adding tokens to the Marian tokenizer to adapt it to new languages proved to be a successful strategy to tackle the problems related to the tokenization of unknown languages.
Both the proposed extensions confirmed the effectiveness of the implemented approach. As a matter of fact, the translation between two low-resource languages with English as an intermediate step provided satisfactory results. Furthermore, the mixed fine-tuning method proved to be useful also on a different dataset, WikiMatrix, which is significantly different from ALT since not all provided examples are reliable (which explains why the final performances are much lower).
Some of the models finetuned in the context of this project are available on the Huggingface hub[15] and can be downloaded and used.

[12] 7.11: mixed + pure
[13] 7.48: mixed 10 epochs
[14] Model trained using Chinese and target language.
[15] https://huggingface.co/CLAck

References

[1] Haddow, Barry & Bawden, Rachel & Miceli Barone, Antonio Valerio & Helcl, Jindřich & Birch, Alexandra. (2021). Survey of Low-Resource Machine Translation.
[2] Zoph, Barret, 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, Association for Computational
[3] Dong, Daxiang, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1723–1732, Association for Computational Linguistics, Beijing, China
[4] Liu, Yinhan, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. Transactions of the Association for Computational Linguistics, 8:726–742.
[5] Dabre, Raj, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting multilingualism through multistage Fine-Tuning for Low-Resource neural machine translation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1410–1416, Hong Kong, China.
[6] Schwenk, Holger & Chaudhary, Vishrav & Sun, Shuo & Gong, Hongyu & Guzman, Francisco. 2019. WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia.
[7] Toral, Antonio and Edman, Lukas and Yeshmagambetova, Galiya and Spenader, Jennifer. 2019. Neural Machine Translation for English–Kazakh with Morphological Segmentation and Synthetic Data