

# Improving Zero-Shot Translation of Low-Resource Languages

Surafel M. Lakew<sup>1,2</sup>, Quintino F. Lotito<sup>2,\*</sup>, Matteo Negri<sup>1</sup>, Marco Turchi<sup>1</sup>, Marcello Federico<sup>1</sup>

<sup>1</sup>Fondazione Bruno Kessler, Trento, Italy

<sup>2</sup>University of Trento, Trento, Italy

federico@fbk.eu

## Abstract

Recent work on multilingual neural machine translation reported competitive performance with respect to bilingual models and surprisingly good performance even on (zero-shot) translation directions not observed at training time. We investigate here a zero-shot translation in a particularly low-resource multilingual setting. We propose a simple iterative training procedure that leverages a duality of translations directly generated by the system for the zero-shot directions. The translations produced by the system (sub-optimal since they contain mixed language from the shared vocabulary), are then used together with the original parallel data to feed and iteratively re-train the multilingual network. Over time, this allows the system to learn from its own generated and increasingly better output. Our approach shows to be effective in improving the two zero-shot directions of our multilingual model. In particular, we observed gains of about 9 BLEU points over a baseline multilingual model and up to 2.08 BLEU over a pivoting mechanism using two bilingual models. Further analysis shows that there is also a slight improvement in the non-zero-shot language directions.

## 1. Introduction

Machine translation of low-resource languages represents a challenge for neural machine translation (NMT) [1]. Recent efforts in multilingual NMT (Multi-NMT) [2, 3] have shown to improve translation performance in low-resource settings. Multi-NMT models can be trained with parallel corpora of several language pairs to work in *many-to-one*, *one-to-many*, or *many-to-many* translation directions. A simple approach, named *target-forcing* [3], is to prepend to the source sentence a tag specifying the target language, both at training and testing time. In addition to performance gains for low-resource languages, the benefit of Multi-NMT is the possibility to perform zero-shot translation, i.e. across directions that were not observed at training time.

Application scenarios in which zero-shot translation can bootstrap the creation of new parallel data – e.g. via human post-editing – [2], show how translation performance in the initial zero-shot direction improves over time with the addition of new parallel data. In this work, we explore instead the

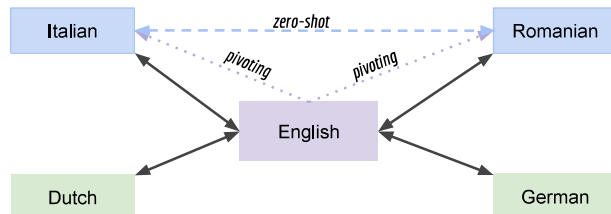


Figure 1: Our zero-shot translation setting for Italian-Romanian. Parallel data is available only for Italian-English, Romanian-English, German-English, and Dutch-English. We leverage multi-lingual neural machine translation trained on all available parallel data to translate across Italian-Romanian (in both directions), either directly (zero-shot) or through English (pivoting).

possibility to enable a trained Multi-NMT model to further learn from its own generated data. Briefly, our method works as follows: first (1), we let the Multi-NMT engine generate zero-shot translations on some portion of the training data; then (2), we re-start the training process on both the generated translations and the original parallel data. We repeat this *training-inference-training* cycle for a few times. Notice that, at each iteration, the original training data is augmented only with the last batch of generated translations. We observe that the generated outputs initially contain a mix of words from the shared vocabulary, but after few iteration they tend to only contain words in the zero-shot target language thus becoming more and more suitable for learning.

We test our approach on a Multi-NMT scenario including Italian, Romanian, English, German and Dutch, assuming that the zero-shot translation pair is Italian-Romanian. We also make the assumption that all languages have just parallel data with English (see Figure 1). We apply our approach on top of the multilingual NMT training method suggested by [2]. Experimental results show that our iterative training procedure not only significantly improves performance on the zero-shot directions, but it also boost multilingual NMT in general. Finally, our approach shows to outperform pivot-based machine translation, too.

\* Work done during a summer internship at FBK.

## 2. Related Work

In this section we discuss relevant works on multilingual NMT, zero-shot NMT, and model training with self-generated data, which are closely related to our approach.

### 2.1. Multilingual NMT

Previous works in Multi-NMT are characterized by the use of separate encoding and/or decoding networks for every translation direction. Dong et al. (2015) [4] proposed a multi-task learning approach for a *one-to-many* translation scenario, based on a sharing representations between related tasks – *i.e* the source language – in order to enhance generalization on the target language. In particular, they used a single encoder in the source side, and separate attention mechanism and decoders for every target language. In a related work [5], used separate encoder and decoder networks for modeling language pairs in a *many-to-many* setting. Notably, they dropped the attention mechanism in favor of a shared vector space where to represent both text and multi-modal information. Aimed at reducing ambiguities at translation time, [6] employed a *multi-source* system that considers two languages on the encoder side and one target language on the decoder side. In particular, the attention model is applied to a combination of the two encoder states. In a *many-to-many* translation scenario, [7] introduced a way to share the attention mechanism across multiple languages. As in [4], but (*only on the decoder side*) and in [5], they used separate encoders and decoders for each source and target language.

Despite the reported improvements, the need of using additional encoder and/or decoder for every language added to the system tells the limitation of these approaches, by making their network complex and expensive to train.

In a very different way, [2] and [3] developed similar Multi-NMT approaches by introducing a *target-forcing* token in the input. The approach in [3] applies a language-specific code to words from different languages in a mixed-language vocabulary. In practice, they force the decoder to translate to a specific target language by prepending and appending an artificial token to the source text. However, their word and sub-word level language-specific coding mechanism significantly increase the input length, which shows to have an impact on the computational cost and performance of NMT [8]. In [2], only one artificial token is prepended to the source sentences in order to specify the target language. Prepending language tokens has permitted to eliminate the need of having separate encoder/decoder networks and attention mechanism for every new language pair.

### 2.2. Zero-Shot Translation

By extending the approach in [7], zero-resource NMT has been suggested in [9]. The authors proposed a *many-to-one* translation setting and used the idea of generating a pseudo-parallel corpus [10], using a pivot language, to fine tune their model. Moreover, also in this case the need of separate en-

coders and decoders for every language pair significantly increases the complexity of the model.

An attractive feature of the *target-forcing* mechanism comes from the possibility to perform zero-shot translation with the same multilingual setting as in [2, 3]. However, recent experiments have shown that the mechanism fails to achieve reasonable zero-shot translation performance for low-resource languages [11]. The promising results in [2] and [3] hence require further investigation to verify if their method can work in various language settings, particularly across distant languages.

### 2.3. Training with self-generated data

Training procedures using self-generated data have been around for a while. For instance, in statistical machine translation (SMT), [12, 13] showed how the output of a translation model can be used iteratively to improve results in a task like post-editing. Mechanisms like back-translating the target side of a single language pair have been used for domain adaptation [14] and more recently by [10] to improve an NMT baseline model. In [15], a dual-learning mechanism is proposed where two NMT models working in the opposite directions provide each other feedback signals that permit them to learn from monolingual data. In a related way, our approach also considers training from monolingual data along dual zero-shot directions. As a difference, however, our *train-infer-train* loop leverages the capability of the network to jointly learn multiple translation directions.

Although our brief survey shows that re-using the output of an MT system for further training and improvement has been successfully applied in different settings, our approach differs from past works in mainly two aspects: *i*) introducing for the first time a *train-infer-train* mechanism addresses Multi-NMT, and *ii*) we cast the approach into a *self-correcting* training procedure over two dual zero-shot directions, so that incrementally improved translations mutually reinforce each direction.

## 3. Neural Machine Translation

The standard NMT architecture comprises an encoder, a decoder and an attention-mechanism, which are all trained with maximum likelihood in an end-to-end fashion [16]. The encoder is a recurrent neural network (RNN) that encodes a source sentence into a sequence of hidden state vectors. The decoder is another RNN that uses the representation of the encoder to predict words in the target language [8] [17]. As the name suggests, *attention* is a mechanism used to improve the translation quality by deciding which part of the source sentence can contribute mostly in the prediction process [18]. As shown in Figure 2, which simplifies the NMT architecture, first the encoder takes the source words on the left (purple color), maps them to vectors and feeds them into the RNN. When the  $\langle \text{eos} \rangle$  (*i.e* end of sentence) symbol is seen, the final time step initializes the decoder RNN (blue

color). At each time step, the attention mechanism is applied over the encoder hidden states and combined with the current hidden state of the decoder to predict the next target word. Then, the prediction is fed back to the encoder RNN to predict the next word, until the  $\langle \text{eos} \rangle$  symbol is generated [19].

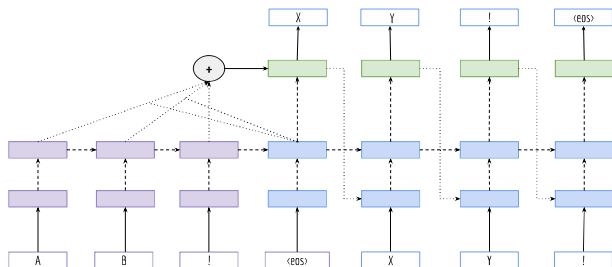


Figure 2: NMT architecture with encoder-decoder and an attention mechanism, showing a source sentence “A B” translated into a target sentence “X Y”.

#### 4. Mixed Language Input for Multi-NMT

Our goal is to improve translation in the zero-shot directions of a multilingual model with limited directions covered by the training data (see Figure 1). The training strategy of the proposed approach is summarized in Algorithm 1, while its flow chart is illustrated in Figure 3.

To address this problem, our training procedure is performed in three steps which are iterated for several rounds. In the first step (line 2), the multilingual NMT system is trained on the original data available. In the second step (line 5), the trained model is run to translate between the zero-shot directions. Then, in the third step (line 8), the output translations are combined with the corresponding source sentences and added to the original training data. The resulting expanded corpus is now ready to perform a new round of the training process.

According to our *train-infer-train* scheme, new *synthetic* data for the two zero-shot directions are generated at each round. This process creates a *duality* between the two zero-shot translation directions, which we can exploit for mutual improvement. Indeed, for each direction, sub-optimal translations  $t^*$  paired with the corresponding original (and well-formed) source  $s$  are used to obtain new “parallel”  $(t^*, s)$  sentence pairs that extend the training material for the other direction. The translated mixed-input for the two languages is represented as  $T^*$ , while the target side  $T$  represents the original sentences extracted for inference.

In the Multi-NMT scenario, the sub-optimal translations representing the source element of the new training pairs will likely contain a mixed-language that includes words from a vocabulary shared with other languages. The expectation is that, round after round, the model will generate better outputs by learning at the same time to translate and “correct” its

---

#### Algorithm 1: Iterative Learning Procedure

---

- 1: TRAIN:  $D(src, tgt)$
  - 2: Multi-NMT  $\leftarrow$  initial training using dataset  $D$
  - 3: **repeat** INFER-TRAIN
  - 4:   **for**  $s = 1, T$  **do**
  - 5:      $t^* \leftarrow$  inference in duality using Multi-NMT
  - 6:   **end for**
  - 7:   prepare  $D^*([src + T^*], [tgt + T])$
  - 8:   Multi-NMT  $\leftarrow$  reload Multi-NMT, train using  $D^*$
  - 9:   return Multi-NMT
  - 10: **until** Multi-NMT converges  $\rightarrow$  Multi-NMT\*
- 

Table 1: *Iterative Learning algorithm of the proposed approach using the duality of zero-shot translation directions.*

own translations by removing spurious elements from other languages. If this intuition holds true, the iterative improvement will yield increasingly better results in translating between the *source* $\leftrightarrow$ *target* zero-shot directions. Ideally, this incremental training and inference cycle can continue until the model converges (line 10).

#### 5. Experiments

All the experiments are carried out using the open source OpenNMT-py<sup>1</sup> toolkit [19]. For training the models, we used the parameters specified in Table 2. Considering the high data sparsity of our low-resource setting, we applied a dropout of 0.3 [20] to prevent overfitting [21]. To train the baseline Multi-NMT, we used Adam [22] as the optimization algorithm with an initial learning rate of 0.001. In the subsequent *train-infer-train* rounds, we used SGD [23], with a learning rate of 1. If the perplexity does not decrease on the validation set or the number of epoch is above 7, a learning rate decay of 0.7 is applied. This combination of optimizers was found to be effective in accelerating the training in the first few iterations. In all the reported experiments the baseline models are trained until convergence, while each train round after the inference stage is assumed to iterate over 10 epochs. For decoding, a beam search of size 10 is applied.

Model parameters	Value
RNN type	LSTM
RNN size	1024
Embedding dim	512
Encoder	bidirectional
Encoder depth	2
Decoder depth	2
Beam size	10
Batch size	128

Table 2: *Hyper-parameters used to train all the models, unless specified in a different setting.*

<sup>1</sup><https://github.com/OpenNMT/OpenNMT-py>

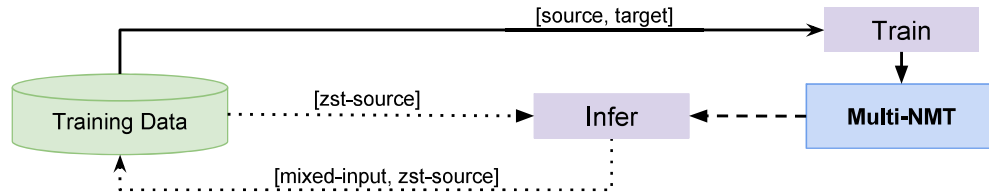


Figure 3: Illustration of the proposed multilingual train-infer-train strategy. Using a standard NMT architecture, a portion of two zero-shot directions monolingual dataset is extracted for inference to construct a dual source $\leftrightarrow$ target mixed-input and continue the training. The top solid line shows the training process, where as the dashed lines show the inference stage

## 5.1. Dataset

To evaluate our approach, we consider five languages (*i.e.* English (EN), Dutch (NL), German (DE), Italian (IT), and Romanian (RO)). To simulate a low-resource scenario, each language pair has  $\approx 200k$  parallel sentences (see Table 3 for details). All the parallel datasets are from the IWSLT17<sup>2</sup> multilingual shared task [24].

Direction	Training	test2010	test2017
EN $\leftrightarrow$ DE	197,489	1,497	1,138
EN $\leftrightarrow$ IT	221,688	1,501	1,147
EN $\leftrightarrow$ NL	231,669	1,726	1,181
EN $\leftrightarrow$ RO	211,508	1,633	1,129
IT $\leftrightarrow$ RO	209,668	1,605	1,127

Table 3: Number of sentences used to train the multilingual model on eight directions. The IT  $\leftrightarrow$  RO pairs are used to train only the bilingual models.

To train all models, we used the same pipeline, first to get a tokenized dataset. Then, we apply byte pair encoding (BPE) [25], using a jointly trained (on source and target dataset) shared BPE model to segment the tokens into sub-word units. For this operation we used 8,000 BPE merging rules, with a minimum of 30 frequency threshold to apply the segmentation. When training the multilingual models, the pipeline includes adding the artificial language token at the source side of each parallel dataset both for the training and validation sets [2]. We evaluate our models using test2010, and for comparison we use test2017 of the IWSLT2017 evaluation dataset.

## 5.2. Models

Our baseline models are trained in a multilingual and bilingual settings. For each direction of the multilingual model and every bilingual model we report the BLEU [26] score computed using *multi-bleu.perl*<sup>3</sup> from the Moses SMT implementation. BLEU scores of the Multi-NMT systems trained on the parallel data in Table 3 are reported in Table 6 and 7 (second column). To com-

<sup>2</sup><https://sites.google.com/site/iwslt2017/>

<sup>3</sup><http://www.statmt.org/moses>

pare our zero-shot translations against those of the bilingual models we trained two Italian $\leftrightarrow$ Romanian models. Both bilingual are trained with the same amount of training data used by each direction of the Multi-NMT model (see Table 3). Moreover, as additional terms of comparison, we trained two pivoting-based systems (using English as a pivot language): Italian $\rightarrow$ English $\rightarrow$ Romanian and Romanian $\rightarrow$ English $\rightarrow$ Italian.

### 5.2.1. Bilingual models

The baseline models for comparison consist of: *i)* an eight direction multilingual model (Multi-NMT), and two bilingual NMT models.

System	tst2010	tst2017
Italian $\rightarrow$ Romanian	19.66	19.14
Romanian $\rightarrow$ Italian	22.44	20.69

Table 4: BLEU scores of two bilingual NMT models (Italian $\rightarrow$ Romanian and Romanian $\rightarrow$ Italian) on IWSLT data *tst2010* and *tst2017*

The results of the two bilingual models are shown in Table 4. From the Multi-NMT model (see row 9 and 10 of Table 6 and Table 7), we particularly focus on the performance of the zero-shot directions that can be compared with the results from these two models.

### 5.2.2. Pivoting

If data are available, the pivoting strategy is the most intuitive way to accomplish zero-shot translation, or to translate from/into under-resourced languages through high resource ones [27]. However, results in the pivoting framework are strictly bounded to the performance of the two combined translation engines, and especially to that of the weaker one. In contrast, Multi-NMT models that leverage knowledge acquired from data for different language combinations (similar to multi-task learning) can potentially compete or even outperform the pivoting ones. Checking this possibility is the motivation for our comparison between the two approaches.

In our experiment we take English as the bridge language between Italian and Romanian in both translation directions.

System	tst2010	tst2017
Italian→Romanian	16.4	15.00
Romanian→Italian	18.9	17.36

Table 5: Performance of the Italian↔Romanian pivot translation directions using English as a pivot on *tst2010* and *tst2017*

Unsurprisingly, compared with those of the bilingual models trained on Italian↔Romanian data, the results shown in Table 5 are lower.

On both translation directions, the bilingual models are indeed about 3.0 BLEU points better. Such comparison, however, is not the main point of our experiment, instead, we aim to fairly analyze performance differences between pivoting and zero-shot methods trained in the same condition which lacks Italian↔Romanian training data.

### 5.3. Zero-shot results

In this experiment, we show how our approach helps to improve the baseline Multi-NMT model. The *train-infer-train* procedure described in Section 4 was applied for five rounds, where each round consists of 10 iterations. Table 6, shows the improvement on the Italian↔Romanian zero-shot directions using the Multi-NMT\* model. Specifically, the Italian→Romanian direction reached 17.38 BLEU score improving over the baseline (8.59) by 8.79 points. Romanian→Italian translation improved with an even larger margin (+10.71) from 8.65 to 19.36 BLEU score.

	Multi-NMT	Multi-NMT*
English→Italian	27.07	<b>28.47</b>
Italian→English	32.12	<b>33.16</b>
English→Romanian	24.65	<b>25.37</b>
Romanian→English	32.7	<b>34.00</b>
English→German	26.39	<b>26.42</b>
German→English	31.3	<b>31.79</b>
English→Dutch	30.27	<b>30.85</b>
Dutch→English	35.13	<b>35.77</b>
Italian→Romanian	8.59	<b>17.38</b>
Romanian→Italian	8.65	<b>19.36</b>

Table 6: Comparison on *test2010* set between a baseline Multi-NMT model against a Multi-NMT\* model with our proposed *train-infer-train* approach for the Italian↔Romanian zero-shot direction. The best result for each direction is shown in bold.

In addition, and to our great surprise, the results from our self-correcting mechanism showed to perform even better than the pivoting strategy. To check the validity of our results, we also compared the baseline multilingual system

	Multi-NMT	Multi-NMT*
English→Italian	29.02	<b>30.43</b>
Italian→English	32.87	<b>33.61</b>
English→Romanian	20.96	<b>21.94</b>
Romanian→English	27.48	<b>28.21</b>
English→German	19.75	<b>19.85</b>
German→English	24.12	<b>24.25</b>
English→Dutch	25.37	<b>26.12</b>
Dutch→English	<b>29.25</b>	29.15
Italian→Romanian	8.18	<b>17.08</b>
Romanian→Italian	8.58	<b>19.25</b>

Table 7: Comparison on *test2017* set between a baseline Multi-NMT model against a Multi-NMT\* model with our proposed *train-infer-train* approach for the Italian↔Romanian zero-shot direction.

and our approach on the IWSLT 2017 test set (*test2017*). As shown in Table 7, the results confirm those computed on *test2010*, with almost identical gains (+8.9 and +10.67). The other important advantage of our approach is evidenced by the performance gains obtained on the language directions supported by parallel training corpora. To put this into perspective, all translation directions have shown improvements, except for the slight drop (-0.10 BLEU) observed for the Dutch→English direction in *test2017* case.

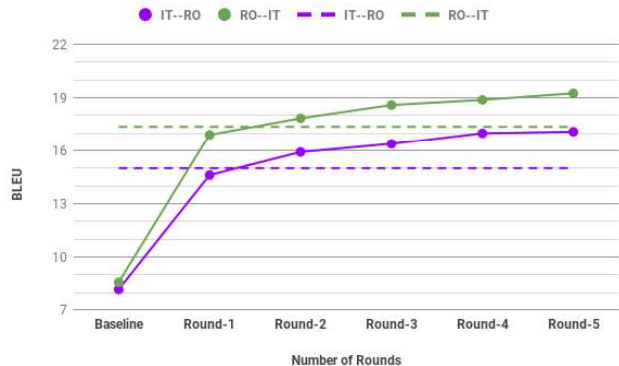


Figure 4: Results from *test2017* for the Italian↔Romanian zero-shot directions, comparing our iterative learning approach (solid lines) with the pivoting mechanism (dashed lines)

Comparing the results from every rounds (see Figure 4), we observe that the training after the first inference step is responsible for the largest portion of the overall gain. This is mainly due to the initial introduction of (noisy) parallel data for the zero-shot directions. The contribution of the self-correcting process can be seen in the following rounds, i.e the improvement after each inference suggests that the generated data are getting cleaner and cleaner.

Italian→Romanian	
<b>Source</b>	... che rafforza la corruzione, l'evasione fiscale, la povertà, l'instabilità.
<b>Pivot</b>	... poarta de bază, evazia fiscală, sărăcia, instabilitatea.
<b>Multi-NMT</b>	... restrânge corupția, <b>fiscale de evasion, poverty, instabilitate.</b>
<b>Multi-NMT*</b>	... care rafinează corupția, <b>evasarea fiscală, sărăcia, instabilitatea.</b>
<b>Reference</b>	... care protejează corupția, evaziunea fiscală, sărăcia și instabilitatea.
Romanian→Italian	
<b>Source</b>	E o poveste incredibilă.
<b>Pivot</b>	È una storia incredibile
<b>Multi-NMT</b>	È una storia <b>incredibile.</b>
<b>Multi-NMT*</b>	È una storia <b>incredibile</b>
<b>Reference</b>	È una storia incredibile .
English→Italian	
<b>Source</b>	We can't use them to make simple images of things out in the Universe.
<b>Multi-NMT</b>	Non possiamo usarli <b>per creare immagini semplici di</b> cose nell'universo.
<b>Multi-NMT*</b>	Non possiamo usarle <b>per fare semplici immagini di</b> cose nell'universo.
<b>Reference</b>	Non possiamo usarle per fare semplici immagini di cose nell'universo

Table 8: Top two examples: zero-shot translations generated by pivoting via English, multilingual translation(Multi-NMT) and multilingual translation enhanced with our approach (Multi-NMT\*). Last example: multilingual and enhanced multi-lingual translation in a resourced translation direction.

Looking at the sample translation outputs using the different approaches in Table 8, we observe that the baseline Multi-NMT system produces mixed language output (e.g. “poverty” in Italian→Romanian and “incredible” in Romanian→Italian). Thanks to our approach, the Multi-NMT\* system instead tends to produce more consistent target language (“poverty” becomes “sărăcia” in Italian→Romanian and “incredible” becomes “incredibile” in Romanian→Italian). Furthermore, even in the non-zero-shot directions there are cases where the enhanced Multi-NMT\* system produces better translations (see the last reported example).

## 6. Conclusions

We introduced a method to improve zero-shot translation in multilingual NMT under particularly resource-scarce training conditions. The proposed self-correcting procedure, by leveraging syntactic dual translations, achieved significant improvements over a multilingual NMT baseline and outperformed a pivoting NMT approach for the Italian-Romanian directions.

In future work, we plan to improve the train-infer-train stages to reach better performance in less time and with lower training complexity. In our current setup we did not consider any form of selection for the dataset to be translated at the inference stage of the train-infer-train procedure. We expect that applying frequency and similarity based approaches to select promising training candidates can bring further improvements. Moreover, we plan also to test our approach with additional monolingual data from the two zero-shot directions. Finally we plan to extend our approach to the translation of mixed language sentences (i.e code-mixing).

## 7. Acknowledgements

This work has been partially supported by the EC-funded projects ModernMT (H2020 grant agreement no. 645487) and QT21 (H2020 grant agreement no. 645452). This work was also supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1 and by a donation of Azure credits by Microsoft. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

## 8. References

- [1] P. Koehn and R. Knowles, “Six challenges for neural machine translation,” *arXiv preprint arXiv:1706.03872*, 2017.
- [2] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, *et al.*, “Google’s multilingual neural machine translation system: Enabling zero-shot translation,” *arXiv preprint arXiv:1611.04558*, 2016.
- [3] T.-L. Ha, J. Niehues, and A. Waibel, “Toward multilingual neural machine translation with universal encoder and decoder,” *arXiv preprint arXiv:1611.04798*, 2016.
- [4] D. Dong, H. Wu, W. He, D. Yu, and H. Wang, “Multi-task learning for multiple language translation.” in *ACL (1)*, 2015, pp. 1723–1732.
- [5] M.-T. Luong, Q. V. Le, I. Sutskever, O. Vinyals, and L. Kaiser, “Multi-task sequence to sequence learning,” *arXiv preprint arXiv:1511.06114*, 2015.



- [6] B. Zoph and K. Knight, “Multi-source neural translation,” *arXiv preprint arXiv:1601.00710*, 2016.
- [7] O. Firat, K. Cho, and Y. Bengio, “Multi-way, multilingual neural machine translation with a shared attention mechanism,” *arXiv preprint arXiv:1601.01073*, 2016.
- [8] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” *arXiv preprint arXiv:1409.1259*, 2014.
- [9] O. Firat, B. Sankaran, Y. Al-Onaizan, F. T. Y. Vural, and K. Cho, “Zero-resource translation with multilingual neural machine translation,” *arXiv preprint arXiv:1606.04164*, 2016.
- [10] R. Sennrich, B. Haddow, and A. Birch, “Improving neural machine translation models with monolingual data,” *arXiv preprint arXiv:1511.06709*, 2015.
- [11] S. M. Lakew, A. D. G. Mattia, and F. Marcelllo, “Multilingual neural machine translation for low resource languages,” in *CLiC-it 2017 – 4th Italian Conference on Computational Linguistics, to appear*, 2017.
- [12] K. Oflazer and I. D. El-Kahlout, “Exploring different representational units in english-to-turkish statistical machine translation,” in *Proceedings of the Second Workshop on Statistical Machine Translation*. Association for Computational Linguistics, 2007, pp. 25–32.
- [13] H. Béchara, Y. Ma, and J. van Genabith, “Statistical post-editing for a statistical mt system,” in *MT Summit*, vol. 13, 2011, pp. 308–315.
- [14] N. Bertoldi and M. Federico, “Domain adaptation for statistical machine translation with monolingual resources,” in *Proceedings of the fourth workshop on statistical machine translation*. Association for Computational Linguistics, 2009, pp. 182–189.
- [15] Y. Xia, D. He, T. Qin, L. Wang, N. Yu, T. Liu, and W. Ma, “Dual learning for machine translation,” *CoRR*, vol. abs/1611.00179, 2016.
- [16] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [17] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [18] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” *arXiv preprint arXiv:1508.04025*, 2015.
- [19] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, “Openmt: Open-source toolkit for neural machine translation,” *arXiv preprint arXiv:1701.02810*, 2017.
- [20] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” in *Advances in neural information processing systems*, 2016, pp. 1019–1027.
- [21] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [22] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [23] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, *et al.*, “Large scale distributed deep networks,” in *Advances in neural information processing systems*, 2012, pp. 1223–1231.
- [24] M. Cettolo, C. Girardi, and M. Federico, “Wit3: Web inventory of transcribed and translated talks,” in *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, vol. 261, 2012, p. 268.
- [25] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” *arXiv preprint arXiv:1508.07909*, 2015.
- [26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.
- [27] H. Wu and H. Wang, “Pivot language approach for phrase-based statistical machine translation,” *Machine Translation*, vol. 21, no. 3, pp. 165–181, 2007.