Multilingual Neural Machine Translation for Low-Resource Languages

Surafel M. Lakew[*]
Fondazione Bruno Kessler
University of Trento

Marcello Federico
MMT Srl, Trento
Fondazione Bruno Kessler

Matteo Negri
Fondazione Bruno Kessler

Marco Turchi
Fondazione Bruno Kessler

Neural Machine Translation (NMT) has been shown to be more effective in translation tasks compared to phrase-based statistical machine translation. However, NMT systems are limited in translating low-resourced languages, due to the significant amount of parallel data that is required to learn useful mappings between languages. In this work, we show how so-called multilingual NMT can help to tackle the challenges associated with low-resourced language translation. The principle of multilingual NMT is to force the creation of hidden representations of words in a shared semantic space across multiple languages, thus enabling a positive parameter transfer across languages. In particular, we present multilingual translation experiments with three languages (English, Italian, Romanian) covering six translation directions, utilizing both recurrent neural networks and transformer (or self-attentive) neural networks. We then focus on the zero-shot translation problem, that is how to leverage multi-lingual data in order to learn translation directions uncovered by the data. Hence, we introduce our recently proposed iterative self-training method, that incrementally improves a multilingual NMT on a zero-shot direction by just relying on monolingual data. Our results on TED talks data show that multilingual NMT outperforms conventional bilingual NMT, that the transformer NMT outperforms recurrent NMT, and that zero-shot NMT outperforms conventional pivoting methods and even matches the performance of a fully-trained bilingual system.

1. Introduction

Neural machine translation (NMT) has shown its effectiveness by delivering the best performance in the IWSLT (Cettolo et al. 2016) and WMT (Bojar et al. 2016) evaluation campaigns of the last three years. Unlike rule-based or statistical MT, the end-to-end learning approach of NMT models the mapping from source to target language directly through a posterior probability. The essential component of an NMT system includes an encoder, a decoder and an attention mechanism (Bahdanau, Cho, and Bengio 2014). Despite the continuous improvement in performance and translation quality (Bentivogli et al. 2018), NMT models are highly dependent on the availability of extensive parallel data, which in practice can only be acquired for a very limited number of language pairs. For this reason, building effective NMT systems for low-resource languages becomes a primary challenge (Koehn and Knowles 2017). In fact, Zoph et al. (2016) showed

[*] Fondazione Bruno Kessler, Via Somarive 18, 38123 Povo (Trento), Italy. E-mail: `lakew@fbk.eu`

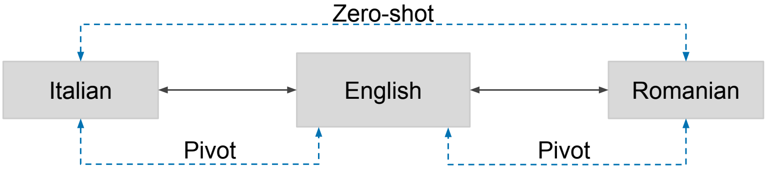

Figure 1
A multilingual setting with parallel training data in four translation directions: Italian↔English and Romanian↔English. Translations between Italian and Romanian are either inferred directly (zero-shot) or by translating through English (pivoting).

how a standard string-to-tree statistical MT system (Galley et al. 2006) can effectively outperform NMT methods for low-resource languages, such as Hausa, Uzbek, and Urdu.

In this work, we approach low-resource machine translation with so-called *multilingual* NMT (Johnson et al. 2016; Ha, Niehues, and Waibel 2016), which considers the use of NMT to target many-to-many translation directions. Our motivation is that intensive cross-lingual transfer (Terence 1989) via parameter sharing across multiple languages should ideally help in the case of similar languages and sparse training data. In particular, we investigate multilingual NMT across Italian, Romanian, and English, and simulate low-resource conditions by limiting the amount of available parallel data.

Among the various approached for multilingual NMT, the simplest and most effective one is to train a single neural network on parallel data including multiple translation directions and to prepend to each source sentence a *flag* specifying the desired target language (Ha, Niehues, and Waibel 2016; Johnson et al. 2016). In this work, we investigate multi-lingual NMT under low-resource conditions and with two popular NMT architectures: recurrent LSTM-based NMT and the self-attentive or transformer NMT model. In particular, we train and evaluate our systems on a collection of TED talks (Cettolo, Girardi, and Federico 2012), over six translation directions: English↔Italian, English↔Romanian, and Italian↔Romanian.

A most relevant benefit that arises from multi-lingual NMT is the possibility of performing a zero-shot translation, that is to query the system on a direction for which no training data was provided. The case we consider is illustrated in Figure 1: we assume to only have Italian-English and English-Romanian training data and that we need to translate between Italian and Romanian in both directions. To solve this task, we propose a *self-learning method* that permits a multi-lingual NMT system trained on the above mentioned language pairs to progressively learn how to translate between Romanian and Italian directly from its own translations. We show that our zero-shot self-training approach not only improves over a conventional pivoting approach, by bridging Romanian-Italian through English (see Figure 1), but that it also matches the performance of bilingual systems trained on Italian-Romanian data. In this work our contribution is two fold[1]:

- Comparing RNN and Transformer approaches in a multilingual setting;

[1] This paper integrates and extends work presented in (Lakew, Di Gangi, and Federico 2017) and (Lakew et al. 2017).

- A self-learning approach to improve the zero-shot translation task of a multilingual model.

This paper is arranged as follows. In section 2, with present previous works on multilingual NMT, zero-shot NMT, and NMT training with self-generated data. In Section 3, we introduce the two prominent NMT approaches evaluated in this paper, the recurrent and the transformer models. In Section 4, we introduce our employed multilingual NMT approach and our self-training method for zero-shot learning. In Section 5, we describe our experimental set-up and NMT model configurations. In Section 6, we present and discuss the results of our experiments. Section 7 ends the paper with conclusions.

2. Previous Work

2.1 Multilingual NMT

Previous works in multilingual NMT are characterized by the use of separate encoding and/or decoding networks for each translation direction. Dong et al. (2015) proposed a multi-task learning approach for a *one-to-many* translation scenario, by sharing hidden representations among related tasks – *i.e* the source languages – to enhance generalization on the target language. In particular, they used a single encoder for all source languages and separate attention mechanisms and decoders for every target language. In a related work (Luong et al. 2015), used distinct encoder and decoder networks for modeling language pairs in a *many-to-many* setting. Aimed at reducing ambiguities at translation time, Zoph and Knight (2016) employed a *many-to-one* 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, Firat, Cho, and Bengio (2016) introduced a way to share the attention mechanism across multiple languages. As in Dong et al. (2015), but (*only on the decoder side*) and in Luong et al. (2015), 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, Johnson et al. (2016) and Ha, Niehues, and Waibel (2016) developed similar multilingual NMT approaches by introducing a *target-forcing* token in the input. The approach in Ha, Niehues, and Waibel (2016) 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 increases the input length, which shows to have an impact on the computational cost and performance of NMT (Cho et al. 2014a). In Johnson et al. (2016), only one artificial token is pre-pended to the entire source sentences in order to specify the target language. Hence, the same token is also used to trigger the decoder generating the translation (cf. Figure 2). Remarkably, pre-pending language tokens to the input string has greatly simplified multi-lingual NMT, by eliminating the need of having separate encoder/decoder networks and attention mechanism for every new language pair.

2.2 Zero-Shot and Self-Learning

Zero-resource NMT has been proposed in Firat et al. (2016) and it extends the work by Firat, Cho, and Bengio (2016). The authors proposed a *many-to-one* translation setting and used the idea of generating a pseudo-parallel corpus (Sennrich, Haddow, and Birch 2015a), using a pivot language, to fine tune their model. However, also in this case the need of separate encoders 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 Johnson et al. (2016) and Ha, Niehues, and Waibel (2016). The both reported that a multilingual system trained on a large amount of data improves over a baseline bilingual model, and that it is also capable of performing zero-shot translation, assuming that the zero-shot source and target languages have been observed during training paired with some other languages.

However, recent experiments have shown that the mechanism fails to achieve reasonable zero-shot translation performance for low-resource languages (Lakew, Di Gangi, and Federico 2017). The promising results in Johnson et al. (2016) and Ha, Niehues, and Waibel (2016) hence require further investigation to verify if their method can work in various language settings, particularly across distant languages.

An alternative approach to zero-shot translation in a resource-scarce scenario is to use a pivot language (Cettolo, Bertoldi, and Federico 2011), that is, using an intermediate language for translation. While this solution is usually pursued by deploying two or more bilingual models, in this work we aim to achieve comparable results using a single multilingual model.

Training procedures using synthetic data have been around for a while. For instance, in statistical machine translation (SMT), Oflazer and El-Kahlout (2007), Béchara, Ma, and van Genabith (2011) 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 (Bertoldi and Federico 2009) and more recently by Sennrich, Haddow, and Birch (2015a) to improve an NMT baseline model. In Xia et al. (2016), 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. As a difference, however, our proposed method leverages the capability of the network to jointly learn multiple translation directions and to directly generate the translations used for self-training.

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 two aspects: *i)* introducing a new self-training method integrated in a multilingual NMT architecture, and *ii)* casting the approach into a *self-correcting* procedure over two dual zero-shot directions, so that incrementally improved translations mutually reinforce each direction.

3. Neural Machine Translation

State-of-the-art NMT systems comprise an encoder, a decoder, and an attention mechanism, which are jointly trained with maximum likelihood in an end-to-end fashion (Bahdanau, Cho, and Bengio 2014). Among the different variants, two popular ones are the recurrent NMT (Sutskever, Vinyals, and Le 2014) and the transformer NMT (Vaswani et

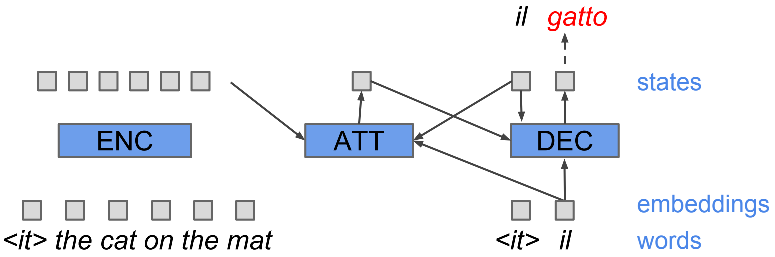

Figure 2
Encoder-decoder-attention NMT architecture. Once the encoder has generated his states for all the input words, the decoder starts generating the translation word by word. The target word "gatto" (cat) is generated from the previously generated target word "il" (the), the previous decoder state and the context state. The context state is a selection and combination of encoder states computed by the attention model. Finally, notice that the target-language forcing symbol "<it>" (Italian), prepended to the input, is also used to trigger the first output word.

al. 2017) models. In both approaches, the encoder is purposed to map a source sentence into a sequence of state vectors, whereas the decoder uses the previous decoder states, its last output, and the attention model state to infer the next target word (see Figure 2). In a broad sense, the attention mechanism selects and combines the encoder states that are mostly relevant to infer the next word (Luong, Pham, and Manning 2015). In our multi-lingual setting, the decoding process is triggered by specifying the target language identifier (Italian, <*it*>, in the example of Figure 2). In the following two subsections, we briefly summarize the main features of the two considered architectures.

3.1 Recurrent NMT

Recurrent NMT models employ recurrent neural networks (RNNs) to build the internal representations of both the encoder and decoder. Recurrent layers are in general implemented with LSTM (Hochreiter and Schmidhuber 1997) or GRU (Cho et al. 2014a) units, which include gates able to control the propagation of information over time. While the encoder typically uses a bi-directional RNN, so that both left-to-right and right-to-left word dependencies are captured (see left-hand of Figure 3), the decoder by design can only learn left-to-right dependencies. In general, deep recurrent NMT is achieved by stacking multiple recurrent layers both inside the encoder and the decoder.

While RNN are in theory the most expressive type of neural networks (Siegelmann and Sontag 1995), they are in practice hard and slow to train. In particular, the combination of two levels of deepness, horizontal along time and vertical across the layers, makes gradient-based optimization of deep RNNs particularly slow to converge and difficult to parallelize (Wu et al. 2016). Recent work succeeded in speeding up training convergence (Sennrich et al. 2017a) of recurrent NMT by reducing the network size via parameter tying and layer normalization. On the other hand the *simple recurrent* NMT model proposed by (Di Gangi and Federico 2018), which weakens the network time dependencies, has shown to outperform LSTM-based NMT both in training speed and performance.

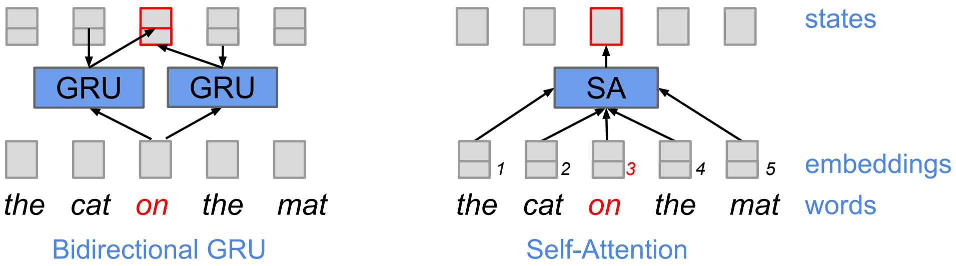

Figure 3
Single-layer encoders with recurrent (left) and transformer networks (right). A bi-directional recurrent encoder generates the state for word "on" with two GRU units. Notice that states must be generated sequentially. The transformer generates the state of word "on" with a self-attention model that looks at all the input embeddings, which are extended with position information. Notice that all the states can be generated independently.

3.2 Transformer NMT

The transformer architecture (Vaswani et al. 2017) works by relying on a self-attention mechanism, removing all the recurrent operations that are found in the RNN case (Vaswani et al. 2017). In other words, the attention mechanism is re-purposed to also compute the latent space representation of both the encoder and the decoder. The right-end side of Figure 3 depicts a simple one-layer encoder based on self-attention. Notice that in absence of recurrence, a *positional-encoding* is added to the input and output embeddings. Similarly, as the time-step in RNN, the positional information provides the transformer network with the order of input and output sequences. In our work we use the absolute positional encoding, but very recently the use of the relative positional information has been shown to improve the network performance (Shaw, Uszkoreit, and Vaswani 2018).

Overall, the transformer is organized as a stack of encoder-decoder networks that works in an auto-regressive way, using the previously generated symbol as input for the next prediction. Both the decoder and encoder can be composed of uniform layers, each built of sub-layers, i.e., a multi-head self-attention sub-layer and a position wise feed-forward network sub-layer. Specifically for the decoder, an extra multi-head attentional layer is added to attend to the output states of the encoder. Multi-head attention layers enable the use of multiple attention functions with a computational cost similar to utilizing a single attention.

4. Zero-Shot Self-Training in Multilingual NMT

In this setting our goal is to improve translation in the zero-shot directions of a baseline multilingual model trained on data covering n languages but not all their possible combinations (see Figure 1). After training a baseline multilingual model with the target-forcing method Johnson et al. (2016), our self-learning approach works in the following way:

- First, a dual zero-shot inference (i.e., source↔target directions) is performed utilizing monolingual data extracted from the training corpus;

- Second, the training resumes combining the inference output and the original multilingual data from the non zero-shot directions;
- Third, this cycle of *training-inference-training* is repeated until a convergence point is reached on the dual zero-shot directions.

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 iterations, they tend to only contain words in the zero-shot target language thus becoming more and more suitable for learning. The training and inference strategy of the proposed approach is summarized in Algorithm 1, whereas the flow chart (see Figure 4) further illustrates the training and inference pipeline.

Table 1
Self-training algorithm for zero-shot directions $l_1 \leftrightarrow l_2$.

Algorithm 1: Train-Infer-Train (TIF)

```
1: TIF: ($D$, $l_1$, $l_2$)
2:   $M \leftarrow$ Train($\emptyset$, $D$)                //train multilingual base model on data $D$
3:   $L_1 \leftarrow$ Extract($D$, $l_1$)              //extract $l_1$ monolingual data from $D$
4:   $L_2 \leftarrow$ Extract($D$, $l_2$)              //extract $l_2$ monolingual data from $D$
5:   for $i = 1, N$ do
6:      $L_2^* \leftarrow$ Infer($M$, $L_1$, $l_2$)         //translate $L_1$ into $l_2$
7:      $L_1^* \leftarrow$ Infer($M^*$, $L_2$, $l_1$)        //translate $L_2$ into $l_1$
8:      $D^* \leftarrow D \cup (L_1^*, L_2) \cup (L_2^*, L_1)$   //augment original data
9:      $M \leftarrow$ Train($M$, $D^*$)             //re-train model on augmented data
10:  end for
11:  return $M$
```

The proposed approach is performed in three steps, where the latter two are iterated for a few rounds. In the first step (line 2), a multilingual NMT system M is trained from scratch on the available data D (*"Train"* step). In the second step (lines 7-8), the last trained model M is run to translate (*"Infer"* step) between the zero-shot directions monolingual data L_1 and L_2 extracted from D (lines 3-4). Then, in the third step (line 10), training of M is re-started on the original data D plus the generated synthetic translations L_2^* and L_1^*, by keeping the extracted monolingual data L_1 and L_2 always on the target side (*"Train"* step). The updated model is then again used to generate synthetic translations, on which to re-train M, and so on.

In the multilingual NMT scenario, the automatic translations used as the source part of the extended training data 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 own translations by removing spurious elements from other languages. If this intuition holds, the iterative improvement will yield increasingly better results in translating between the source $\leftrightarrow$ target zero-shot directions.

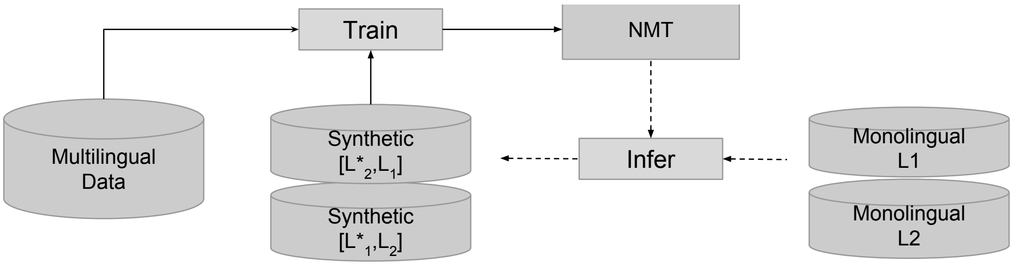

Figure 4
Illustration of the proposed multilingual train-infer-train *strategy. Using a standard NMT architecture, a portion of two zero-shot directions monolingual dataset is extracted for inference to construct a dual source↔target mixed-input and continue the training. The solid lines show the training process, whereas the dashed lines indicate the inference stage.*

5. Experiments

5.1 NMT Settings

We trained and evaluated multilingual NMT systems based on the RNN (Cho et al. 2014b) and transformer (Vaswani et al. 2017) models. Table 2 summarizes the hyper-parameters used for all our models. The RNN experiments are carried out using the NMT toolkit Nematus[2] (Sennrich et al. 2017b), whereas the transformer models are trained using the open source OpenNMT-tf[3] toolkit (Klein et al. 2017).

Training and inference hyper-parameters for both approaches and toolkits are fixed as follows. For the RNN experiments, the Adagrad (Duchi, Hazan, and Singer 2011) optimization algorithm is utilized with an initial learning rate of 0.01 and mini-batches of size 100. Considering the high data sparsity of our low-resource setting and to prevent model over-fitting (Srivastava et al. 2014) we applied a dropout on every layer, with probability 0.2 on the embeddings and the hidden layers, and 0.1 on the input and output layers. For the experiments using the transformer approach, a dropout of 0.3 is used globally. To train the baseline multilingual NMT, we use Adam (Kingma and Ba 2014) as the optimization algorithm with an initial learning rate scale constant of 2. For the transformer, the learning rate is increased linearly in the early stages (*warmup_training_steps*=16000); after that, it is decreased with an inverse square root of training step (Vaswani et al. 2017).

In all the reported experiments the baseline models are trained until convergence, while each training round after the inference stage is assumed to iterate for 3 epochs. In case of the transformer NMT, M4-NMT, and M6-NMT scores are computed using averaged model from the last seven checkpoints in same training run (Junczys-Dowmunt, Dwojak, and Sennrich 2016). For decoding, a beam search of size 10 is applied for recurrent models, whereas one of size 4 is used for transformer models.

[2] https://github.com/EdinburghNLP/nematus
[3] https://github.com/OpenNMT/OpenNMT-tf

Table 2
Hyper-parameters used to train RNN and transformer models, unless differently specified.

	enc/dec type	embedding size	hidden units	encoder depth	decoder depth	batch size
RNN	GRU	1024	1024	2	2	128 seg
Transformer	Transformer	512	512	6	6	4096 tok

5.2 Dataset and preprocessing

We run all our experiments on the multilingual translation shared task prepared by the 2017 International Workshop on Spoken Language Translation (IWSLT)[4]. In particular, we used the subset of training data covering all possible language pair combinations between Italian, Romanian, and English (Cettolo, Girardi, and Federico 2012). For development and evaluating the models, we used the corresponding sets from the IWSLT2010 (Paul, Federico, and Stüker 2010) and IWSLT2017 evaluation campaigns. Details about the used data sets are reported in Table 3. At the preprocessing stage, we applied word segmentation for each training condition (i.e. bilingual or multi-lingual) by learning a sub-word dictionary via Byte-Pair Encoding (Sennrich, Haddow, and Birch 2015b), by setting the number of merging rules to 39,500. We observe a rather high overlap between the language pairs (i.e the English dataset paired with Romanian is highly similar to the English paired with Italian). Because of this overlapping, the actual unique sentences in the dataset are approximately half of the total size. Consequently, on one side, this exacerbates the low-resource aspect in the multilingual models, while, on the other side we expect some positive effect on the zero-shot condition. The final size of the vocabulary both in case of the bilingual and the multilingual models stays under 40,000 sub-words. An evaluation script to compute the BLEU (Papineni et al. 2002) score is used to validate models on the dev set and later to choose the best performing models. Furthermore, significance tests computed for the BLEU are reported using Multeval (Clark et al. 2011).

Table 3
The total number of parallel sentences used for training, development, and test in our low-resource scenario.

Language Pair	*Train*	*Test10*	*Test17*
En-It	231619	929	1147
En-Ro	220538	929	1129
It-Ro	217551	914	1127

We trained models for two different scenarios. The first is the multi-lingual scenario with all the available language pairs, while the second is for the zero-shot and pivoting approaches which excludes $Romanian - Italian$ parallel sentences from the training data. For both scenarios, we have also trained bilingual *RNN* and *Transformer* models

[4] http://workshop2017.iwslt.org/

for comparing bilingual against multilingual systems and for comparing pivoting with bilingual and multilingual models.

6. Models and Results

6.1 Bilingual Vs. Multilingual NMT

We compare the translation performance of six independently trained bilingual models against one single multilingual model trained on the concatenation of all the six language pairs datasets, after prepending the language flag on the source side of each sentence. The performance of both types of systems is evaluated on test2017 and reported in Table 4. The experiments show that a multilingual system is either favorably compares to or outperforms the bilingual systems. Improvements are observed in all language directions, which are likely gained from the cross-lingual parameter transfer between the additional language pairs involved in the source and target side.

Table 4
Comparison between six bilingual models (NMT) against a single multilingual model (M6-NMT) on Test17.

Direction	*RNN*			*Transformer*		
	NMT	M6-NMT	Δ	NMT	M6-NMT	Δ
En → It	27.44	28.22	+0.78	29.24	30.88	+1.64
It → En	29.9	31.84	+1.94	33.12	36.05	+2.93
En → Ro	20.96	21.56	+0.60	23.05	24.65	+1.60
Ro → En	25.44	27.24	+1.80	28.40	30.25	+1.85
It → Ro	17.7	18.95	+1.25	20.10	20.13	+0.03
Ro → It	19.99	20.72	+0.73	21.36	21.81	+0.45

Table 4 shows that the transformer model is definitely superior than the RNN model for all directions and set-ups. With a larger margin of +3.22 (NMT) and +4.21 (M6-NMT) the transformer outperforms the RNN in the It-En direction. The closest performance between the two approaches is observed in the Ro-It direction, with the transformer showing a +1.37 (NMT) and +1.09 (M6-NMT) better BLEU scores compared to the RNN counter part. Moreover, multilingual architectures in general outperforms their equivalent models trained on single language pairs. The highest improvement of the M6-NMT over the NMT systems is observed when the target language is English. For instance in the It-En direction the multilingual approach gained +1.94 (RNN) and +2.93 (Transformer) over the single language pair models. Similarly a +1.80 (RNN) and +1.85 (Transformer) gains are observed in the Ro-En direction. However, the smallest gain of the multilingual models occurred when translating to either Italian or Romanian. Despite the instance of target languages in the experimental setting, the slight difference in the dataset size (that tends to benefit the English target, see Table 3) showed to impact the performance on non English target directions.

6.2 Pivoting using a Multilingual Model

The pivoting experiment is setup by dropping the Italian↔Romanian parallel segments from the training data, and by training *i)* a new multilingual-model covering four directions and *ii) a single model for each language direction (It → En, En → It, Ro → En, En → Ro)*. Our main aim is to analyze how a multilingual model can improve a zero-shot translation task using a pivoting mechanism with English as a bridge language in the experiment. Moreover, the use of a multilingual model for pivoting is motivated by the results we acquired using the M6-NMT (see Table 4).

Table 5
Comparison of pivoting with bilingual models (NMT) and with multilingual models (M4-NMT) on Test17.

Direction	*RNN*			*Transformer*		
	NMT	M4-NMT	$\Delta^{\texttt{M4-NMT}}_{\texttt{NMT}}$	NMT	M4-NMT	$\Delta^{\texttt{M4-NMT}}_{\texttt{NMT}}$
It → En → Ro	16.3	17.58	+1.28	16.59	16.77	+0.18
Ro → En → It	18.69	18.66	-0.03	17.87	19.39	+1.52

The results in Table 5 show the potential, although partial, of using multilingual models for pivoting unseen translation directions. The comparable results achieved in both directions speak to us in favor of training and deploying one system instead of two distinct NMT systems. Remarkably, the marked difference between RNN and transformer is vanished in this condition. Pivoting using M4-NMT system showed to perform better in three out of four evaluations, from the RNN and transformer runs. Note that the performance of the final translation (i.e pivot-target) is subject to the noise that has been propagated from the source-pivot translation step. Meaning pivoting is a favorable strategy when we have strong models in both directions of the pivot language.

6.3 Zero-shot Translations

For the direct zero-shot experiments and the application of the *train-infer-train* strategy, we only carried out experiments with the transformer approach. Preliminary results showed its superiority over the RNN together with the possibility to carry out experiments faster and with multiple GPUs.

In this experiment, we show how our approach helps to significantly boost the baseline multilingual NMT model. We run the train-infer-train for five consecutive stages, where each round consist in 2-3 epochs of additional training on the augmented training data. Table 6 shows the improvements on the dual Italian↔Romanian zero-shot directions.

Table 6
Comparison between a baseline multilingual model (M4-NMT) against the results from our proposed *train-infer-train* approach in a subsequent five rounds for the Italian↔Romanian zero-shot directions.

Direction	*M4-NMT*	*R1*	*R2*	*R3*	*R4*	*R5*
It→Ro	4.72	15.22	18.46	19.31	19.59	**20.11**
Ro→It	5.09	16.31	20.31	21.44	21.63	**22.41**

In both zero-shot directions the gain in a larger margin comes using the M-NMT model at *R1*. This is the first model trained after the inclusion of the dataset generated by the *dual-inference* stage. The It→Ro direction improves by $+10.50$ BLEU points from a 4.72 to 15.22, whereas Ro→It improved from a baseline score of 5.09 to 16.31 BLEU ($+11.22$).

The contribution of the self correcting process can be seen in the subsequent rounds, i.e., the improvements after each inference stage suggests that the generated data are getting cleaner and cleaner. With respect to the Transformer model, pivoting results shown in Table 5, our approach outperformed both single pair and multilingual pivoting methods at the second round (*R2*) (see the third column of Table 6). Compared with the better performing multilingual pivoting, our approach at the fifth round (*R5*) has a $+3.34$ and $+3.02$ BLEU gain for the It→Ro and Ro→It directions respectively.

Table 7
Results summary comparing the performance of systems trained using parallel data (i.e., two single language pair *NMT* and a six direction multilingual *M6-NMT* systems) against the four directions multilingual baseline (*M4-NMT*) and our approach at the fifth round *R5*. Best scores are bold highlighted, whereas statistically significant (p<0.05) results in comparison with the baseline (*NMT*) are indicated with $\star$

Direction	*NMT*	*M6-NMT*	$\Delta_{\texttt{NMT}}^{\texttt{M6-NMT}}$	*M4-NMT*	$\Delta_{\texttt{NMT}}^{\texttt{M4-NMT}}$	*R5*	$\Delta_{\texttt{NMT}}^{\texttt{R5}}$
It→Ro	20.10	**20.13**	+0.03	4.72	-15.38	20.11	+0.01
Ro→It	21.36	21.81	+0.04	5.09	-16.27	**22.41***	+1.05

In addition to outperforming the pivoting mechanism, an interesting trend arises when we compare our approach with the results of the single language pair and multilingual models reported in Table 4. The summary in Table 7 shows how effective a dual-inference mechanism in allowing the model to learn from its outputs. Compared to the models trained using parallel data (i.e., *NMT* and *M6-NMT*), our approach (*R5*) is either comparable ($+0.01$ BLEU in It→Ro) or better performing ($+1.05$ BLEU in Ro→It). Overall our iterative self-learning approach showed to deliver better results than bilingual counterparts within five rounds, where each rounds iterates for a maximum of three epochs. Indeed the improvement from our approach is a concrete example to train models in a self-learning way, potentially benefiting language directions with a parallel data, if casted in a similar setting.

7. Conclusions

In this paper, we used a multilingual NMT model in a low-resource language pairs scenario. We showed that a single multilingual system outperforms bilingual baselines while avoiding the need to train several single language pair models. In particular, we confirmed the superiority of transformer over recurrent NMT architectures in a multilingual setting. For enabling and improving a zero-shot translation we showed i) how a multilingual pivoting can be used for achieving comparable results to those of multiple bilingual models, and ii) that our proposed self-learning procedure boost performance of multilingual zero-shot directions by even outperforming both pivoting and bilingual models.

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.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Béchara, Hanna, Yanjun Ma, and Josef van Genabith. 2011. Statistical post-editing for a statistical mt system. In *MT Summit*, volume 13, pages 308–315.

Bentivogli, Luisa, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2018. Neural versus phrase-based mt quality: An in-depth analysis on english-german and english-french. *Computer Speech & Language*, 49:52–70.

Bertoldi, Nicola and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the fourth workshop on statistical machine translation*, pages 182–189. Association for Computational Linguistics.

Bojar, Ondrej, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 conference on machine translation (wmt16). In *Proceedings of the First Conference on Machine Translation (WMT)*, volume 2, pages 131–198.

Cettolo, Mauro, Nicola Bertoldi, and Marcello Federico. 2011. Bootstrapping arabic-italian smt through comparable texts and pivot translation. In *15th Annual Conference of the European Association for Machine Translation (EAMT)*.

Cettolo, Mauro, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.

Cettolo, Mauro, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2016. The iwslt 2016 evaluation campaign. *Proc. of IWSLT, Seattle, pp. 14, WA, 2016*.

Cho, Kyunghyun, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Clark, Jonathan H, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 176–181. Association for Computational Linguistics.

Di Gangi, Mattia A. and Marcello Federico. 2018. Deep neural machine translation with weakly-recurrent units. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation (EAMT)*, Alicante, Spain, May. European Association for Machine Translation.

Dong, Daxiang, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *ACL (1)*, pages 1723–1732.

Duchi, John, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Firat, Orhan, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. *arXiv preprint arXiv:1601.01073*.

Firat, Orhan, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. *arXiv preprint arXiv:1606.04164*.

Galley, Michel, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 961–968. Association for Computational Linguistics.

Ha, Thanh-Le, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Johnson, Melvin, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Junczys-Dowmunt, Marcin, Tomasz Dwojak, and Rico Sennrich. 2016. The amu-uedin submission to the wmt16 news translation task: Attention-based nmt models as feature functions in phrase-based smt. *arXiv preprint arXiv:1605.04809*.

Kingma, Diederik and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Koehn, Philipp and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Lakew, S. Melaku, Mattia Antonino Di Gangi, and Marcello Federico. 2017. Multilingual neural machine translation for low resource languages. In *CLiC-it 2017 âĂŞ 4th Italian Conference on Computational Linguistics, to appear*.

Lakew, Surafel M, Quintino F Lotito, Negri Matteo, Turchi Marco, and Federico Marcello. 2017. Improving zero-shot translation of low-resource languages. In *14th International Workshop on Spoken Language Translation*.

Luong, Minh-Thang, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Luong, Minh-Thang, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Oflazer, Kemal and Ilknur Durgar El-Kahlout. 2007. Exploring different representational units in english-to-turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32. Association for Computational Linguistics.

Papineni, Kishore, 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.

Paul, Michael, Marcello Federico, and Sebastian Stüker. 2010. Overview of the iwslt 2010 evaluation campaign. In *International Workshop on Spoken Language Translation (IWSLT) 2010*.

Sennrich, Rico, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017a. The university of edinburgh's neural mt systems for wmt17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark, September. Association for Computational Linguistics.

Sennrich, Rico, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, et al. 2017b. Nematus: a toolkit for neural machine translation. *arXiv preprint arXiv:1703.04357*.

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

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

Shaw, Peter, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Siegelmann, H.T. and E.D. Sontag. 1995. On the computational power of neural nets. *Journal of Computer and System Sciences*, 50(1):132 – 150.

Srivastava, Nitish, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Terence, Odlin. 1989. Language transfer-cross-linguistic influence in language learning. *Cambridge University Press. Cambridge Books Online.*, page 222, June.

Vaswani, Ashish, 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 6000–6010.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xia, Yingce, Di He, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. *CoRR*, abs/1611.00179.

Zoph, Barret and Kevin Knight. 2016. Multi-source neural translation. *arXiv preprint arXiv:1601.00710*.

Zoph, Barret, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.