Extremely low-resource neural machine translation for Asian languages

Raphael Rubino¹ · Benjamin Marie¹ · Raj Dabre¹ · Atushi Fujita¹ · Masao Utiyama¹ · Eiichiro Sumita¹

Received: 4 March 2020 / Accepted: 24 December 2020 / Published online: 10 February 2021
© The Author(s) 2021

Abstract

This paper presents a set of effective approaches to handle extremely low-resource language pairs for self-attention based neural machine translation (NMT) focusing on English and four Asian languages. Starting from an initial set of parallel sentences used to train bilingual baseline models, we introduce additional monolingual corpora and data processing techniques to improve translation quality. We describe a series of best practices and empirically validate the methods through an evaluation conducted on eight translation directions, based on state-of-the-art NMT approaches such as hyper-parameter search, data augmentation with forward and backward translation in combination with tags and noise, as well as joint multilingual training. Experiments show that the commonly used default architecture of self-attention NMT models does not reach the best results, validating previous work on the importance of hyper-parameter tuning. Additionally, empirical results indicate the amount of synthetic data required to efficiently increase the parameters of the models leading to the best translation quality measured by automatic metrics. We show that the best NMT models trained on large amount of tagged back-translations outperform three other synthetic data generation approaches. Finally, comparison with statistical machine translation (SMT) indicates that extremely low-resource NMT requires a large amount of synthetic parallel data obtained with back-translation in order to close the performance gap with the preceding SMT approach.

Keywords Neural machine translation · Low-resource · Asian language · Transformer · Synthetic data · Hyper-parameter tuning

✉ Raphael Rubino
raphael.rubino@nict.go.jp

Atushi Fujita
atsushi.fujita@nict.go.jp

¹ ASTREC, NICT, 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan

1 Introduction

Neural machine translation (NMT) (Forcada and Neco 1997; Cho et al. 2014; Sutskever et al. 2014; Bahdanau et al. 2015) is nowadays the most popular machine translation (MT) approach, following two decades of predominant use by the community of statistical MT (SMT) (Brown et al. 1991; Och and Ney 2000; Zens et al. 2002), as indicated by the majority of MT methods employed in shared tasks during the last few years (Bojar et al. 2016; Cettolo et al. 2016). A recent and widely-adopted NMT architecture, the so-called Transformer (Vaswani et al. 2017), has been shown to outperform other models in high-resource scenarios (Barrault et al. 2019). However, in low-resource settings, NMT approaches in general and Transformer models in particular are known to be difficult to train and usually fail to converge towards a solution which outperforms traditional SMT. Only a few previous work tackle this issue, whether using NMT with recurrent neural networks (RNNs) for low-resource language pairs (Sennrich and Zhang 2019), or for distant languages with the Transformer but using several thousands of sentence pairs (Nguyen and Salazar 2019).

In this paper, we present a thorough investigation of training Transformer NMT models in an extremely low-resource scenario for distant language pairs involving eight translation directions. More precisely, we aim to develop strong baseline NMT systems by optimizing hyper-parameters proper to the Transformer architecture, which relies heavily on multi-head attention mechanisms and feed-forward layers. These two elements compose a usual *Transformer* block. While the number of blocks and the model dimensionality are hyper-parameters to be optimized based on the training data, they are commonly fixed following the model topology introduced in the original implementation.

Following the evaluation of baseline models after an exhaustive hyper-parameter search, we provide a comprehensive view of synthetic data generation making use of the best performing tuned models and following several prevalent techniques. Particularly, the tried-and-tested back-translation method (Sennrich et al. 2016a) is explored, including the use of source side tags and noised variants, as well as the forward translation approach.

The characteristics of the dataset used in our experiments allow for multilingual (many-to-many languages) NMT systems to be trained and compared to bilingual (one-to-one) NMT models. We propose to train two models: using English on the source side and four Asian languages on the target side, or vice-versa (one-to-many and many-to-one). This is achieved by using a data pre-processing approach involving the introduction of tags on the source side to specify the language of the aligned target sentences. By doing so, we leave the Transformer components identical to one-way models to provide a fair comparison with baseline models.

Finally, because monolingual corpora are widely available for many languages, including the ones presented in this study, we contrast low-resource supervised NMT models trained and tuned as baselines to unsupervised NMT making use of a large amount of monolingual data. The latter being a recent alternative to low-resource NMT, we present a broad illustration of available state-of-the-art techniques making use of publicly available corpora and tools.

To the best of our knowledge, this is the first study involving eight translation directions using the Transformer in an extremely low-resource setting with less than 20k parallel sentences available. We compare the results obtained with our NMT models to SMT, with and without large language models trained on monolingual corpora. Our experiments deliver a *recipe* to follow when training Transformer models with scarce parallel corpora.

The remainder of this paper is organized as follows. Section 2 presents previous work on low-resource MT, multilingual NMT, as well as details about the Transformer architecture and its hyper-parameters. Section 3 introduces the experimental settings, including the datasets and tools used, and the training protocol of the MT systems. Section 4 details the experiments and obtained results following the different approaches investigated in our work, followed by an in-depth analysis in Sect. 5. Finally, Sect. 6 gives some conclusions.

2 Background

This section presents previous work on low-resource MT, multilingual NMT, followed by the technical details of the Transformer model and its hyper-parameters.

2.1 Low-resource MT

NMT systems usually require a large amount of parallel data for training. Koehn and Knowles (2017) conducted a case study on English-to-Spanish translation task and showed that NMT significantly underperforms SMT when trained on less than 100 million words. However, this experiment was performed with standard hyper-parameters that are typical for high-resource language pairs. Sennrich and Zhang (2019) demonstrated that better translation performance is attainable by performing hyper-parameter tuning when using the same attention-based RNN architecture as in Koehn and Knowles (2017). The authors reported higher BLEU scores than SMT on simulated low-resource experiments in English-to-German MT. Both studies have in common that they did not explore realistic low-resource scenarios nor distant language pairs leaving unanswered the question of the applicability of their findings to these conditions.

In contrast to parallel data, large quantity of monolingual data can be easily collected for many languages. Previous work proposed several ways to take advantage of monolingual data in order to improve translation models trained on parallel data, such as a separately trained language model to be integrated to the NMT system architecture (Gulcehre et al. 2017; Stahlberg et al. 2018) or exploiting monolingual data to create synthetic parallel data. The latter was first proposed for SMT (Schwenk 2008; Lambert et al. 2011), but remains the most prevalent one in NMT (Barrault et al. 2019) due to its simplicity and effectiveness. The core idea is to use an existing MT system to translate monolingual data to produce a sentence-aligned parallel corpus, whose source or target is synthetic. This corpus is then mixed to the initial, non-synthetic, parallel training data to retrain a new MT system.

Nonetheless, this approach only leads to slight improvements in translation quality for SMT. It has been later exploited in NMT (Sennrich et al. 2016a) using the original monolingual data on the target side and the corresponding generated translations on the source side to retrain NMT. A large body of subsequent work proposed to improve this approach that we denote in this paper “backward translation.” In particular, Edunov et al. (2018) and Caswell et al. (2019) respectively showed that adding synthetic noise or a tag to the source side of the synthetic parallel data effectively enables the exploitation of very large monolingual data in high-resource settings. In contrast, Burlot and Yvon (2018) showed that the quality of the backward translations has a significant impact for training good NMT systems.

As opposed to backward translation, Imamura and Sumita (2018) proposed to perform “forward translation”¹ to enhance NMT by augmenting their training data with synthetic data generated by MT on the target side. They only observed moderate improvements, but in contrast to backward translation, this approach does not require a pre-existing NMT system trained for the reverse translation direction.

Recent work has also shown remarkable results in training MT systems using only monolingual data in so-called unsupervised statistical (USMT) and neural (UNMT) machine translation (Artetxe et al. 2018; Lample et al. 2018; Artetxe et al. 2018; Lample et al. 2018). These approaches have only worked in configurations where we do not necessarily need them, i.e., in high-resource scenarios and/or for very close language pairs. Marie et al. (2019) showed that USMT can almost reach the translation quality of supervised NMT for close language pairs (Spanish+6h–Portuguese) while Marie et al. (2019) showed that this approach is unable to generate a translation for distant language pairs (English–Gujarati and English–Kazakh).

To the best of our knowledge, none of these approaches have been successfully applied to NMT systems for extremely low-resource language pairs, such as Asian languages that we experimented with in this paper.

2.2 Multilingual NMT

Most NMT architectures are flexible enough to incorporate multiple translation directions, through orthogonal approaches involving data pre-processing or architecture modifications. One such modification is to create separate encoders and decoders for each source and target language and couple them with a shared attention mechanism (Firat et al. 2016). However, this method requires substantial NMT architecture modifications and leads to an increase in learnable model parameters, which rapidly becomes computationally expensive. At the data level, a straightforward and powerful approach was proposed by Johnson et al. (2017), where a standard NMT model was used as a black-box and an artificial tokens was added on the source side of the parallel data. Such a token could be for instance $\langle 2_{xx} \rangle$, where xx indicates the target language.

¹ This is denoted “self-training” in their paper.

During training, the NMT system uses these source tokens to produce the designated target language and thus allows to translate between multiple language pairs. This approach allows for the so-called *zero-shot* translation, enabling translation directions that are unseen during training and for which no parallel training data is available (Firat et al. 2016; Johnson et al. 2017).

One of the advantages of multilingual NMT is its ability to leverage high-resource language pairs to improve the translation quality on low-resource ones. Previous studies have shown that jointly learning low-resource and high-resource pairs leads to improved translation quality for the low-resource one (Firat et al. 2016; Johnson et al. 2017; Dabre et al. 2019). Furthermore, the performance tends to improve as the number of language pairs (consequently the training data) increases (Aharoni et al. 2019).

However, to the best of our knowledge, the impact of joint multilingual NMT training with the Transformer architecture in an extremely low-resource scenario for distant languages through the use of a multi-parallel corpus was not investigated previously. This motivates our work on multilingual NMT, which we contrast with bilingual models through an exhaustive hyper-parameter tuning. All the multilingual NMT models trained and evaluated in our work are based on the approach presented in Johnson et al. (2017), which relies on the artificial token indicating the target language and prepended onto the source side of each parallel sentence pair.

2.3 Transformer

Encoder–decoder models consist in encoding an input sequence $\mathbf{X} = \{x_1, x_2, \dots, x_n\}$ ($\mathbf{X} \in \mathbb{R}^{n \times d}$) and producing a corresponding output sequence $\mathbf{Y} = \{y_1, y_2, \dots, y_m\}$ ($\mathbf{Y} \in \mathbb{R}^{m \times d}$), where d is the model dimensionality and n and m are the input and output sequence lengths, respectively. The Transformer is built on stacked L_e encoder and L_d decoder layers, each layer consisting of sub-layers of concatenated multi-head dot-product scaled attention (Eq. 1) and two position-wise feed-forward layers with a non-linear activation function in between, usually a rectified linear unit (ReLU) (Nair and Hinton 2010) (Eq. 2).

$$\text{attn}(\mathbf{t}_1, \mathbf{t}_2) = [\text{head}_1; \dots; \text{head}_i; \dots; \text{head}_h] \mathbf{W}^O, \quad (1)$$

$$\text{head}_i(\mathbf{t}_1, \mathbf{t}_2) = \text{softmax}\left(\frac{\mathbf{QW}_i^Q \cdot (\mathbf{KW}_i^K)^T}{\sqrt{d}}\right) \mathbf{VW}_i^V, \quad (2)$$

$$\text{ffn}(\mathbf{H}_{\text{attn}}^l) = \text{ReLU}(\mathbf{H}_{\text{attn}}^l \mathbf{W}_1 + b_1) \mathbf{W}_2 + b_2.$$

Both encoder and decoder layers contain self-attention and feed-forward sub-layers, while the decoder contains an additional encoder–decoder attention sub-layer.² The hidden representations produced by self-attention and feed-forward sub-layers in the

² For more details about the self-attention components, please refer to Vaswani et al. (2017).

l_e -th encoder layer ($l_e \in \{1, \dots, L_e\}$) are formalized by $\mathbf{H}_{Eattn}^{l_e}$ (Eq. 3) and $\mathbf{H}_{Effn}^{l_e}$ (Eq. 4), respectively. Equivalently, the decoder sub-layers in the l_d -th decoder layer ($l_d \in \{1, \dots, L_d\}$) are formalized by $\mathbf{H}_{Dattn}^{l_d}$ (Eq. 5), $\mathbf{H}_{EDattn}^{l_d}$ (Eq. 6), and $\mathbf{H}_{Dffn}^{l_d}$ (Eq. 7). The sub-layer $\mathbf{H}_{Eattn}^{l_e}$ for $l_e = 1$ receives the input sequence embeddings $\mathbf{X}$ instead of $\mathbf{H}_{Effn}^{l_e-1}$, as it is the first encoder layer. The same applies to the decoder.

$$\mathbf{H}_{Eattn}^{l_e} = \nu \left(\chi \left(\mathbf{H}_{Effn}^{l_e-1}, \text{attn} \left(\mathbf{H}_{Effn}^{l_e-1}, \mathbf{H}_{Effn}^{l_e-1} \right) \right) \right), \quad (3)$$

$$\mathbf{H}_{Effn}^{l_e} = \nu \left(\chi \left(\mathbf{H}_{Eattn}^{l_e}, \text{ffn} \left(\mathbf{H}_{Eattn}^{l_e} \right) \right) \right), \quad (4)$$

$$\mathbf{H}_{Dattn}^{l_d} = \nu \left(\chi \left(\mathbf{H}_{Dffn}^{l_d-1}, \text{attn} \left(\mathbf{H}_{Dffn}^{l_d-1}, \mathbf{H}_{Dffn}^{l_d-1} \right) \right) \right), \quad (5)$$

$$\mathbf{H}_{EDattn}^{l_d} = \nu \left(\chi \left(\mathbf{H}_{Dattn}^{l_d}, \text{attn} \left(\mathbf{H}_{Dattn}^{l_d}, \mathbf{H}_{Effn}^{L_e} \right) \right) \right), \quad (6)$$

$$\mathbf{H}_{Dffn}^{l_d} = \nu \left(\chi \left(\mathbf{H}_{EDattn}^{l_d}, \text{ffn} \left(\mathbf{H}_{EDattn}^{l_d} \right) \right) \right). \quad (7)$$

Each sub-layer includes *LayerNorm* (Ba et al. 2016), noted ν , parameterized by $\mathbf{g}$ and $\mathbf{b}$, with input vector $\hat{\mathbf{h}}$, mean μ , and standard deviation φ (Eq. 8), and a residual connection noted χ with input vectors $\hat{\mathbf{h}}$ and $\hat{\mathbf{h}}'$ (Eq. 9). In the default Transformer architecture, *LayerNorm* is placed after each non-linearity and residual connection, also called *post-norm*. However, an alternative configuration is *pre-norm*, placing the normalization layer prior to the non-linearity.

$$\nu(\hat{\mathbf{h}}) = \frac{\hat{\mathbf{h}} - \mu}{\varphi} \odot \mathbf{g} + \mathbf{b} \quad (8)$$

$$\chi(\hat{\mathbf{h}}, \hat{\mathbf{h}}') = \hat{\mathbf{h}} + \hat{\mathbf{h}}' \quad (9)$$

2.4 Hyper-parameters

The Transformer architecture has a set of hyper-parameters to be optimized given the training data. Based on the formalism introduced in Sect. 2.3, the architecture-dependent hyper-parameters to optimize are the model dimensionality d , which is equal to the input and output token embedding dimensionality, the number of heads h , the size of the feed-forward layers *ffn* and thus the dimensions of the parameter matrices $\mathbf{W}_1$ and $\mathbf{W}_2$, as well as the biases b_1 and b_2 , and the number of encoder and decoder layers L_e and L_d , respectively. In addition, it was shown in recent work that the position of the normalization layer ν , before or after the attention or feed-forward layers, leads to training instability (i.e., no

convergence) for deep Transformer architectures or in low-resource settings using an out-of-the-box hyper-parameter configuration (Wang et al. 2019; Nguyen and Salazar 2019).

Our exhaustive hyper-parameter search for NMT models is motivated by the findings of Sennrich and Zhang (2019) where the authors conducted experiments in low-resource settings using RNNs and showed that commonly used hyper-parameters do not lead to the best results. However, the impact of various number of layers was not evaluated in their study. This limitation could lead to sub-optimal translation quality, as other studies on mixed or low-resource settings have opted for a reduced number of layers in the Transformer architecture, for instance using 5 encoder and decoder layers instead of the usual 6 (Schwenk et al. 2019; Chen et al. 2019). In these latter publications, authors also used fewer number of attention heads, between 2 and 4, compared to the out-of-the-box 8 heads from the vanilla Transformer.

Moreover, a number of recent studies on the Transformer architecture have shown that not all attention heads are necessary. For instance, Voita et al. (2019) evaluated the importance of each head in the multi-head attention mechanism in a layer-wise fashion. They identified their lexical and syntactic roles and proposed a head pruning mechanism. However, their method is only applied to a fully-trained 8-head model. A concurrent study carried out by Michel et al. (2019), both for NMT and natural language inference tasks, focused on measuring the importance of each head through a masking approach. This method showed that a high redundancy is present in the parameters of most heads given the rest of the model and that some heads can be pruned regardless of the test set used. An extension to their approach allows for head masking during the training procedure. Results indicate that the importance of each head is established by the Transformer at an early stage during NMT training.

Finding the best performing architecture based on a validation set given a training corpus is possible through hyper-parameter grid-search or by relying on neural architecture search methods (c.f. Elsken et al. (2018)). The former solution is realistic in our extremely low-resource scenario and the latter is beyond the scope of our work.

3 Experimental settings

The aim of the study presented in this paper is to provide a set of techniques which tackles low-resource related issues encountered when training NMT models and more specifically Transformers. This particular NMT architecture involves a large amount of hyper-parameter combinations to be tuned, which is the cornerstone of building strong baselines. The focus of our work is a set of extremely low-resource distant language pairs with a realistic data setting, based on corpora presented in Sect. 3.1. We compare traditional SMT models to supervised and unsupervised NMT models with the Transformer architecture, whose specificities and training procedures are presented in Sect. 3.2. Details about the post-processing and evaluation methods are finally presented in Sect. 3.3.

Table 1 Statistics of the parallel data used in our experiments

	Train	Validation	Test
Sentences	18,088	1000	1018
English			
Tokens	412,995	22,293	22,937
Types	51,084	7230	7305
Japanese			
Tokens	1,088,441	58,609	60,480
Types	2605	1587	1597
Lao			
Tokens	2,218,046	118,448	123,431
Types	202	140	138
Malay			
Tokens	380,854	20,608	21,310
Types	44,337	6421	6570
Vietnamese			
Tokens	577,781	31,262	32,228
Types	30,131	4899	5185

Tokens and types for Japanese and Lao are calculated at the character level

3.1 Dataset

The dataset used in our experiments contains parallel and monolingual data. The former composes the training set of the baseline NMT systems, as well as the validation and test sets, while the latter constitutes the corpora used to produce synthetic parallel data, namely backward and forward translations.

3.1.1 Parallel corpus

The parallel training, validation, and test sets were extracted from the Asian Language Treebank (ALT) corpus (Riza et al. 2016).³ We focus on four Asian languages, i.e., Japanese, Lao, Malay, and Vietnamese, aligned to English, leading to eight translation directions. The ALT corpus comprises a total of 20, 106 sentences initially taken from the English Wikinews and translated into the other languages. Thus, the English side of the corpus is considered as *original* while the other languages are considered as *translationese* (Gellerstam 1986), i.e., texts that shares a set of lexical, syntactic and/or textual features distinguishing them from non-translated texts. Statistics for the parallel data used in our experiments are presented in Table 1.

³ <http://www2.nict.go.jp/astrec-att/member/mutiyaama/ALT/>.

Table 2 Statistics for the entire monolingual corpora, comprising 163M, 87M, 737k, 15M, and 169M lines, respectively for English, Japanese, Lao, Malay, and, Vietnamese, the sampled sub-corpora made of 18k (18,088), 100k, 1M, 10M, or 50M lines. Tokens and types for Japanese and Lao are calculated at the character level

	18k	100k	1M	10M	50M	All
English						
Tokens	276k	1,534k	15,563k	157,909k	793,916k	2,586,016k
Types	47k	145k	664k	3,149k	9,622k	22,339k
Japanese						
Tokens	654k	3,633k	36,299k	362,714k	1,813,162k	3,142,982k
Types	3k	4k	6k	9k	12k	13k
Lao						
Tokens	2125k	11,707k	–	–	–	86,213k
Types	425	655	–	–	–	2k
Malay						
Tokens	243k	1,341k	13,439k	134,308k	–	205,922k
Types	46k	139k	601k	2,533k	–	3,286k
Vietnamese						
Tokens	414k	2,289k	22,876k	228,916k	1,144,380k	3,869,717k
Types	32k	94k	417k	1,969k	5,824k	12,930k

3.1.2 Monolingual corpus

We used monolingual data provided by the Common Crawl project,⁴ which were crawled from various websites in any language. We extracted the data from the April 2018 and April 2019 dumps. To identify from the dumps the lines in English, Japanese, Lao, Malay, and Vietnamese, we used the fastText (Bojanowski et al. 2016)⁵ pretrained model for language identification.⁶ The resulting monolingual corpora still contained a large portion of noisy data, such as long sequences of numbers and/or punctuation marks. For cleaning, we decided to remove lines in the corpora that fulfill at least one of the following conditions:

- more than 25% of its tokens are numbers or punctuation marks.⁷

⁴ <https://commoncrawl.org/>.

⁵ <https://github.com/facebookresearch/fastText/>, version 0.9.1.

⁶ We used the large model presented in <https://fasttext.cc/blog/2017/10/02/blog-post.html>.

⁷ For the sake of simplicity, we only considered tokens that contain at least one Arabic numeral or one punctuation marks referenced in the Python3 constant “string.punctuation”.

Table 3 Hyper-parameters considered during the tuning of baseline NMT systems

Symbol	Description	Values
Training		
<i>d</i>	Embedding dimensions	512, 1024
<i>enc</i>	Number of encoder layers	1, 2, 4, 6
<i>dec</i>	Number of decoder layers	1, 2, 4, 6
<i>h</i>	Number of attention heads	1, 4, 8
<i>ff</i>	Feed-forward dimensions	512, 2048, 4096
<i>norm</i>	Position of normalization	pre- or post-non-linearity
Decoding		
<i>beam</i>	Decoder beam size	1, 4, 12
<i>len</i>	Decoder length penalty	0.6, 1.0, 1.4

- contains less than 4 tokens.⁸
- contains more than 150 tokens.

Statistics of the monolingual corpora and their sub-samples used in our experiments are presented in Table 2.

3.2 MT systems

We trained and evaluated three types of MT systems: supervised NMT, unsupervised NMT, and SMT systems. The computing architecture at our disposal used for our NMT systems consists in 8 Nvidia Tesla V100 with the CUDA library version 10.2. Each of the following sections gives the details of each type of MT systems, including tools, pre-processing, hyper-parameters, and training procedures used.

3.2.1 Supervised NMT systems

Our supervised NMT systems include ordinary bilingual (one-to-one) and multilingual (one-to-many and many-to-one) NMT systems. All the systems were trained using the *fairseq* toolkit (Ott et al. 2019) based on PyTorch.⁹

The only pre-processing applied to the parallel and monolingual data used for our supervised NMT systems was a sub-word transformation method (Sennrich et al. 2016b). Neither tokenization nor case alteration was conducted in order to keep our method language agnostic as much as possible. Because Japanese and Lao languages do not contain spacing, we employed a sub-word transformation approach,

⁸ For English, Malay, and Vietnamese, we counted the tokens returned by the *Moses* tokenizer with the option “-l en” while for Japanese and Lao we counted the tokens returned by *sentencepiece* (Kudo and Richardson 2018). Note that while we refer these tokenized results for this filtering purpose, for training and evaluating our MT systems, we applied dedicated pre-processing for each type of MT systems.

⁹ Python version 3.7.5, PyTorch version 1.3.1.

sentencepiece (Kudo and Richardson 2018), which is based on sequences of characters. Models were learned on joint vocabularies for bilingual NMT systems and on all languages for the multilingual ones. When the same script is used between languages, the shared vocabulary covers all the observed sub-words of these languages. When different scripts are used between languages, only common elements (punctuation, numbers, etc.) are shared. We restricted the number of sub-word transformation operations to 8000 for all the bilingual NMT models and to 32,000 for all the multilingual NMT models.

Our main objective is to explore a vast hyper-parameter space to obtain our baseline supervised NMT systems, focusing on specific aspects of the Transformer architecture. Our choice of hyper-parameters (motivated in Sect. 2.4) and their values are presented in Table 3. These are based on our preliminary experiments and on the findings of previous work that showed which Transformer hyper-parameters have the largest impact on translation quality measured by automatic metrics (Sennrich and Zhang 2019; Nguyen and Salazar 2019). The exhaustive search of hyper-parameter combinations resulted in 576 systems trained and evaluated for each translation direction, using the training and validation sets specific to each language pair presented in Table 1. Because the experimental setting presented in this paper focuses on extremely low-resource languages, it is possible to train and evaluate a large number of NMT systems without requiring too much computing resources.

We trained all the supervised NMT systems following the same pre-determined procedure, based on gradient descent using the Adam optimizer (Kingma and Ba 2014) and the cross-entropy objective with smoothed labels based on a smoothing rate of 0.1. The parameters of the optimizer were $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$. The learning rate was scheduled as in Vaswani et al. (2017), initialized at 1.7^{-7} and following an initial 4k steps warmup before decaying at the inverse squared root rate. The NMT architecture used the ReLU (Nair and Hinton 2010) activation function for the feed-forward layers and scaled dot-product attention for the self and encoder–decoder attention layers. A dropout rate of 0.1 was applied to all configurations during training and no gradient clipping was used. We used a batch of 1, 024 tokens and stopped training after 200 epochs for the baseline systems, 80 epochs for the systems using additional synthetic data and 40 epochs for the largest (10M) data configurations. Bilingual models were evaluated every epoch, while multilingual models were evaluated every 5 epochs, both using BLEU on non post-processed validation sets (including sub-words). The best scoring models were kept for final evaluation.

For each trained model, we introduced two additional hyper-parameters for decoding, i.e., the decoder beam size and translation length penalty as shown in Table 3, to be tuned. The best combination of these two parameters were kept to decode the test.

3.2.2 Unsupervised NMT systems

Our UNMT systems used the Transformer-based architecture proposed by Lample et al. (2018). This architecture relies on a denoising autoencoder as language model during training, on a latent representation shared across languages for the encoder

and the decoder, and pre-initialized cross-lingual word embeddings. To set up a state-of-the-art UNMT system, we used for initialization a pre-trained cross-lingual language model, i.e., XLM, as performed by Lample and Conneau (2019).

UNMT was exclusively trained on monolingual data. For Lao and Malay we used the entire monolingual data, while we randomly sampled 50 million lines from each of the English, Japanese, and Vietnamese monolingual data. To select the best models, XLM and then UNMT models, we relied on the same validation sets used by our supervised NMT systems. Since pre-training an XLM model followed by the training of an UNMT system is costly, we chose to train a single multilingual XLM model and UNMT system for all translation directions. To construct the vocabulary of the XLM model, we concatenated all the monolingual data and trained a *sentencepiece* model with 32,000 operations. The *sentencepiece* model was then applied to the monolingual data for each language and we fixed the vocabulary size during training at 50,000 tokens.

For training XLM and UNMT, we ran the framework publicly released by Lample and Conneau (2019),¹⁰ with default parameters, namely: 6 encoder and decoder layers, 8 heads, 1024 embedding and 4096 feed-forward dimensions, layer dropout and attention dropout of 0.1, and the GELU activation function. Note that XLM and UNMT models must use the same hyper-parameters and that we could not tune these hyper-parameters due to the prohibitive cost of training. Our XLM model is trained using the masked language model objective (MLM) which is usually trained monolingually. However, since we have trained a single model using the monolingual data for all the languages with a shared vocabulary, our XLM model is cross-lingual. The training steps for the denoising autoencoder component were language specific while the back-translation training steps combined all translation directions involving English.

3.2.3 SMT systems

We used *Moses* (Koehn et al. 2007)¹¹ and its default parameters to conduct SMT experiments, including MSD lexicalized reordering models and a distortion limit of 6. We learned the word alignments with *mgiza* for extracting the phrase tables. We used 4-gram language models trained, without pruning, with *LMPLZ* from the *kenlm* toolkit (Heafield et al. 2013) on the entire monolingual data. For tuning, we used *kb-mira* (Cherry and Foster 2012) and selected the best set of weights according to the BLEU score obtained on the validation data after 15 iterations.

In contrast to our NMT systems, we did not apply *sentencepiece* on the data for our experiments with SMT. Instead, to follow a standard configuration typically used for SMT, we tokenized the data using *Moses* tokenizer for English, Malay, and Vietnamese, only specifying the language option “-l en” for these three languages. For Japanese, we tokenized our data with *MeCab*¹² while we used an in-house tokenizer for Lao.

¹⁰ <https://github.com/facebookresearch/XLM>.

¹¹ <https://github.com/moses-smt/mosesdecoder>.

¹² <https://taku910.github.io/mecab/>.

3.3 Post-processing and evaluation

Prior to evaluating NMT system outputs, the removal of spacing and special characters introduced by *sentencepiece* was applied to translations as a single post-processing step. For the SMT system outputs, detokenization was conducted using the *detokenizer.perl* tool included in the *Moses* toolkit for English, Malay and Vietnamese languages, while simple space deletion was applied for Japanese and Lao.

We measured the quality of translations using two automatic metrics, BLEU (Papineni et al. 2002) and chrF (Popović 2015) implemented in SacreBLEU (Post 2018) and relying on the reference translation available in the ALT corpus. For target languages which do not contain spaces, i.e., Japanese and Lao, the evaluation was conducted using the chrF metric only, while we used both chrF and BLEU for target languages containing spaces, i.e., English, Malay and Vietnamese. As two automatic metrics are used for hyper-parameter tuning of NMT systems for six translation directions, the priority is given to models performing best according to chrF, while BLEU is used as a tie-breaker. Systems outputs comparison and statistical significance tests are conducted based on bootstrap resampling using 500 iterations and 1000 samples, following (Koehn 2004).

4 Experiments and results

This section presents the experiments conducted in extremely low-resource settings using the Transformer NMT architecture. First, the baseline systems resulting from exhaustive hyper-parameter search are detailed. Second, the use of monolingual data to produce synthetic data is introduced and two setups, backward and forward translation, are presented. Finally, two multilingual settings, English-to-many and many-to-English, are evaluated.

4.1 Baselines

The best NMT architectures and decoder hyper-parameters were determined based on the validation set and, in order to be consistent for all translation directions, on the chrF metric. The final evaluation of best architectures were conducted by translating the test set, whose evaluation scores are presented in the following paragraphs and summarized in Table 4.¹³ The best configurations according to automatic metrics were kept for further experiments on each translation direction using monolingual data (presented in Sect. 4.2).

¹³ All the results obtained during hyper-parameter search are submitted as supplementary material to this paper.

Table 4 Test set results obtained with our SMT and NMT systems before and after hyper-parameter tuning (NMT_v and NMT_t, respectively), along with the corresponding architectures, for the eight translation directions evaluated in our baseline experiments. Tuned parameters are: *d* model dimension, *ff* feed-forward dimension, *enc* encoder layers, *dec* decoder layers, *h* heads, *norm* normalization position. Results in bold indicate systems significantly better than the others with $p < 0.05$

	EN → JA		JA → EN		EN → LO		LO → EN		EN → MS		MS → EN		EN → VI		VI → EN	
	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU
NMT _v	0.042	0.2	0.130	0.131	0.4	0.170	0.213	1.7	0.194	0.9	0.134	0.6	0.195	0.195	0.195	0.195
NMT _t	0.212	8.6	0.372	0.339	10.5	0.374	0.605	29.9	0.559	26.8	0.468	21.9	0.484	0.484	0.484	0.484
SMT	0.240	9.8	0.401	0.358	13.2	0.418	0.637	31.1	0.585	30.9	0.509	23.9	0.516	0.516	0.516	0.516
Best NMT _t configurations																
<i>d</i>	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512
<i>ff</i>	512	512	512	512	512	512	512	512	512	512	512	4096	512	512	512	512
<i>enc</i>	2	2	2	1	4	4	6	4	4	4	4	4	4	4	4	4
<i>dec</i>	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
<i>h</i>	1	1	1	4	1	1	1	4	4	4	4	4	4	4	4	4
<i>norm</i>	Post	Post	Post	Post	Pre	Pre	Pre	Post	Post	Pre	Post	Pre	Pre	Post	Post	Post
<i>beam</i>	12	12	12	12	12	12	4	12	12	12	12	12	12	12	12	12
<i>len</i>	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4	1.4

4.1.1 English-to-Japanese

Vanilla Transformer architecture (*base* configuration with embeddings of 512 dimensions, 6 encoder and decoder layers, 8 heads, 2048 feed-forward dimensions, post-normalization) with default decoder parameters (beam size of 4 and length penalty set at 0.6) leads to a chrF score of 0.042, while the best configuration after hyper-parameters tuning reaches 0.212 (+0.170pts). The final configuration has embeddings of 512 dimensions, 2 encoder and 6 decoder layers with 1 attention head, 512 feed-forward dimensions with post-normalization. For decoding, a beam size of 12 and a length penalty set at 1.4 lead to the best results on the validation set.

4.1.2 Japanese-to-English

Out-of-the-box Transformer architecture leads to a BLEU of 0.2 and a chrF of 0.130. The best configuration reaches 8.6 BLEU and 0.372 chrF (+8.4pts and +0.242pts respectively). The best configuration according to the validation set has 512 embeddings dimensions, 2 encoder and 6 decoder layers with 1 attention heads, 512 feed-forward dimensions and post-normalization. The decoder parameters are set to a beam size of 12 and a length penalty of 1.4.

4.1.3 English-to-Lao

Default architecture reaches a chrF score of 0.131 for this translation direction. After hyper-parameter search, the score reaches 0.339 chrF (+0.208 pts) with a best architecture composed of 1 encoder and 6 decoder layers with 4 heads, 512 dimensions for both embeddings and feed-forward, using post-normalization. The decoder beam size is 12 and the length penalty is 1.4.

4.1.4 Lao-to-English

Without hyper-parameter search, the results obtained on the test set are 0.4 BLEU and 0.170 chrF. Tuning leads to 10.5 (+10.1pts) and 0.374 (+0.204pts) for BLEU and chrF respectively. The best architecture has 512 dimensions for both embeddings and feed-forward, pre-normalization, 4 encoder and 6 decoder layers with 1 attention head. A beam size of 12 and a length penalty of 1.4 are used for decoding.

4.1.5 English-to-Malay

For this translation direction, the default setup leads to 0.9 BLEU and 0.213 chrF. After tuning, with a beam size of 4 and a length penalty of 1.4 for the decoder, a BLEU of 33.3 (+32.4pts) and a chrF of 0.605 (+0.392pts) are reached. These scores are obtained

with the following configuration: embedding and feed-forward dimensions of 512, 6 encoder and decoder layers with 1 head and pre-normalization.

4.1.6 Malay-to-English

Default Transformer configuration reaches a BLEU of 1.7 and a chrF of 0.194. The best configuration found during hyper-parameter search leads to 29.9 BLEU (+28.2pts) and 0.559 chrF (+0.365pts), with both embedding and feed-forward dimensions at 512, 4 encoder and 6 decoder layers with 4 heads and post-normalization. For decoding, a beam size of 12 and a length penalty set at 1.4 lead to the best results on the validation set.

4.1.7 English-to-Vietnamese

With no parameter search for this translation direction, 0.9 BLEU and 0.134 chrF are obtained. With tuning, the best configuration reaches 26.8 BLEU (+25.9pts) and 0.468 chrF (+0.334pts) using 512 embedding dimensions, 4, 096 feed-forward dimensions, 4 encoder and decoder layers with 4 heads and pre-normalization. The decoder beam size is 12 and the length penalty is 1.4.

4.1.8 Vietnamese-to-English

Using default Transformer architecture leads to 0.6 BLEU and 0.195 chrF. After tuning, 21.9 BLEU and 0.484 chrF are obtained with a configuration using 512 dimensions for embeddings and feed-forward, 4 encoder and 6 decoder layers with 4 attention heads, using post-normalization. A beam size of 12 and a length penalty of 1.4 are used for decoding.

4.1.9 General observations

For all translation directions, the out-of-the-box Transformer architecture (NMT_v) does not lead to the best results and models trained with this configuration fail to converge. Models with tuned hyper-parameters (NMT_t) improve over NMT_v for all translation directions, while SMT remains the best performing approach. Among the two model dimensionalities (hyper-parameter noted d) evaluated, the smaller one leads to the best results for all language pairs. For the decoder hyper-parameters, the length penalty set at 1.4 is leading to the best results on the validation set for all language pairs and translation directions. A beam size of 12 is the best performing configuration for all pairs except for EN→MS, with a beam size of 4. However, for this particular translation direction, only the chrF score is lower when using a beam size of 12 while the BLEU score is identical.

Table 5 Test set results obtained when using backward-translated monolingual data in addition to the parallel training data. Baseline NMT₁ uses only parallel training data and baseline SMT uses monolingual data for its language model

	EN → JA		JA → EN		EN → LO		LO → EN		EN → MS		MS → EN		EN → VI		VI → EN	
	chrF		BLEU	chrF	chrF		BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF
Baselines																
NMT ₁	0.212		8.6	0.372	0.339		10.5	0.374	33.3	0.605	29.4	0.554	26.8	0.468	21.9	0.484
SMT	0.243		11.7	0.415	0.355		16.1	0.436	38.3	0.649	36.5	0.619	33.8	0.527	28.3	0.548
Best baseline architecture (NMT₁) with backward translation																
18k	0.191		8.8	0.361	0.314		9.3	0.359	32.4	0.596	28.6	0.547	26.9	0.469	20.4	0.471
100k	0.202		10.6	0.388	0.339		12.8	0.396	34.2	0.615	32.5	0.581	29.1	0.488	24.6	0.507
1M	0.201		11.1	0.401	0.349		14.9	0.427	36.2	0.629	36.2	0.611	31.5	0.506	28.4	0.538
10M	0.166		10.7	0.390	–		15.2	0.427	35.3	0.625	38.9	0.629	31.8	0.512	30.2	0.553
Transformer base architecture (NMT₁) with backward translation																
18k	0.175		8.9	0.356	0.323		10.2	0.365	30.9	0.590	28.5	0.546	24.4	0.448	20.6	0.470
100k	0.203		11.0	0.394	0.341		13.3	0.397	35.2	0.621	32.5	0.579	29.2	0.489	24.8	0.505
1M	0.240		14.0	0.437	0.364		15.8	0.433	36.8	0.635	36.8	0.614	31.6	0.508	28.1	0.537
10M	0.231		14.5	0.444	–		17.4	0.455	38.1	0.646	39.7	0.638	32.9	0.519	30.7	0.558

4.2 Monolingual data

This section presents the experiments conducted in adding synthetic parallel data to our baseline NMT systems using monolingual corpora described in Sect. 3.1. Four approaches were investigated, including three based on backward translation where monolingual corpora used were in the target language, and one based on forward translation where monolingual corpora used were in the source language. As backward translation variations, we investigated the use of a specific tag indicating the origin of the data, as well as the introduction of noise into the synthetic data.

4.2.1 Backward translation

The use of monolingual corpora to produce synthetic parallel data through backward translation (or back-translation, i.e., translating from target to source language) has been popularized by Sennrich et al. (2016a). We made use of the monolingual data presented in Table 2 and the best NMT systems presented in Table 4 (NMT_t) to produce additional training data for each translation direction.

We evaluated the impact of different amounts of synthetic data, i.e., 18k, 100k, 1M and 10M, as well as two NMT configurations per language direction: the best performing baseline as presented in Table 4 (NMT_t) and the out-of-the-box Transformer configuration, henceforth Transformer *base*¹⁴, noted NMT_v. Comparison of the two architectures along with different amounts of synthetic data allows us to evaluate how much backward translations are required in order to switch back to the commonly used Transformer configuration. These findings are illustrated in Sect. 5.

A summary of the best results involving back-translation are presented in Table 5. When comparing the best tuned baseline (NMT_t) to the Transformer *base* (NMT_v) with 10M monolingual data, NMT_v outperforms NMT_t for all translation directions. For three translation directions, i.e., EN → JA, EN → MS and EN → VI, SMT outperforms NMT architectures. For the EN → LO direction, however, NMT_v outperforms SMT, which is explained by the small amount of Lao monolingual data which does not allow for a robust language model to be used by SMT.

4.2.2 Tagged backward translation

Caswell et al. (2019) empirically showed that adding a unique token at the beginning of each backward translation, i.e., on the source side, acts as a tag that helps the system during training to differentiate backward translations from the original parallel training data. According to the authors, this method is as effective as introducing synthetic noise for improving translation quality (Edunov et al. 2018) (noised backward translation is investigated in Sect. 4.2.3). We believe that the tagged approach is simpler than the noised one since it requires only one editing operation, i.e., the addition of the tag.t

¹⁴ The base model and its hyper-parameters as presented in Vaswani et al. (2017).

Table 6 Test set results obtained when using tagged backward-translated monolingual data in addition to the parallel training data. Baseline NMT₁ uses only parallel training data and baseline SMT uses monolingual data for its language model. Results in bold indicate systems significantly better than the others with $p < 0.05$

	EN → JA		JA → EN		EN → LO		LO → EN		EN → MS		MS → EN		EN → VI		VI → EN	
	chrF		BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF
Baselines																
NMT ₁	0.212		8.6	0.372	0.339		10.5	0.374	33.3	0.605	29.4	0.554	26.8	0.468	21.9	0.484
SMT	0.243		11.7	0.415	0.355		16.1	0.436	38.3	0.649	36.5	0.619	33.8	0.527	28.3	0.548
Best baseline architecture (NMT₁) with tagged backward translation																
18k	0.197		8.4	0.359	0.326		9.9	0.362	32.3	0.597	29.2	0.551	25.7	0.461	21.0	0.473
100k	0.216		10.2	0.390	0.350		13.1	0.406	35.2	0.620	32.0	0.579	28.6	0.484	24.9	0.511
1M	0.223		12.7	0.422	0.369		15.7	0.434	37.9	0.642	36.2	0.615	33.9	0.525	29.0	0.546
10M	0.191		12.1	0.413	–		15.8	0.437	37.2	0.638	39.0	0.632	34.3	0.528	31.0	0.558
Transformer base architecture (NMT₀) with tagged backward translation																
18k	0.177		8.8	0.355	0.326		10.5	0.368	31.9	0.594	28.7	0.545	25.2	0.458	20.6	0.469
100k	0.213		11.8	0.402	0.348		13.4	0.407	35.9	0.627	32.8	0.580	29.1	0.493	24.0	0.504
1M	0.253		16.1	0.460	0.382		16.4	0.445	38.8	0.651	37.8	0.621	33.9	0.526	29.1	0.547
10M	0.253		17.5	0.471	–		18.4	0.459	40.5	0.662	40.1	0.639	35.2	0.535	32.0	0.568

Table 7 Test set results obtained when using noised backward-translated monolingual data in addition to the parallel training data. The NMT architecture is the Transformer *base* configuration. Baseline NMT_t uses only parallel training data and baseline SMT uses monolingual data for its language model

	EN → JA		JA → EN		EN → LO		LO → EN		EN → MS		MS → EN		EN → VI		VI → EN	
	chrF		BLEU	chrF	chrF		BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF
Baselines																
NMT _t	0.212		8.6	0.372	0.339		10.5	0.374	33.3	0.605	29.4	0.554	26.8	0.468	21.9	0.484
SMT	0.243		11.7	0.415	0.355		16.1	0.436	38.3	0.649	36.5	0.619	33.8	0.527	28.3	0.548
Transformer base architecture (NMT_v) with noised backward translation																
18k	0.172		7.9	0.343	0.305		9.9	0.360	29.5	0.578	25.2	0.523	23.7	0.443	19.2	0.454
100k	0.189		10.4	0.381	0.332		12.7	0.396	33.3	0.612	29.7	0.560	27.7	0.479	23.5	0.496
1M	0.234		12.7	0.429	0.367		15.8	0.431	37.7	0.639	36.1	0.610	32.2	0.513	27.7	0.535
10M	0.233		14.6	0.445	-		17.4	0.448	37.4	0.643	39.5	0.633	33.4	0.522	30.5	0.556

To study the effect of tagging backward translation in an extremely low-resource configuration, we performed experiments with the same backward translations used in Sect. 4.2.1 modified by the addition of a tag “[BT]” at the beginning of each source sentence on the source side of the synthetic parallel data. Table 6 presents the results obtained with our tagged back-translation experiments along with the best tuned NMT baseline using only parallel data (NMT_p) and an SMT system using the monolingual data for its language model.

The tagged back-translation results indicate that the Transformer *base* architecture trained in this way outperforms all other approaches for the eight translation directions. It also improves over the backward translation approach without tag (see Table 5), according to both automatic metrics. All translation directions benefit from adding 10M backward translations except for the EN → JA direction, for which we observe a plateau and no improvement over the system using 1M synthetic parallel data.

4.2.3 Noised backward translation

As an alternative to tagged backward translation, we propose to evaluate the noised backward translation approach for the best performing configuration obtained in the previous section, namely the Transformer *base* architecture. Adding noise to the source side of backward translated data has previously been explored in NMT and we followed the exact approach proposed by Edunov et al. (2018). Three types of noise were added to each source sentence: word deletion with a probability of 0.1, word replacement by a specific token following a probability of 0.1, and finally word swapping with random swap of words no further than three positions apart.

Our experiments with noisy backward-translated source sentences were conducted using the Transformer *base* architecture only, as this configuration was leading to the best results when using tagged back-translation. The results are presented in Table 7 and show that tagged backward-translation reaches higher scores for both automatic metrics in all translation directions. Systems trained on noised backward translation outperforms our SMT system except for the EN → JA, EN → MS, and EN → VI translation directions, which are translation directions involving translationese as target.

4.2.4 Forward translation

In contrast to backward translation, with forward translation we used the synthetic data produced by NMT on the target side. Basically, we obtained this configuration by reversing the training parallel data used to train NMT with backward translations in Sect. 4.2.1. One advantage of this approach is that we have clean and original data on the source side to train a better encoder, while a major drawback is that we have synthetic data on the target side, which potentially coerces the decoder into generating ungrammatical translations. Bogoychev and Sennrich (2019) showed that forward translation is more useful when translating from original texts compared to translating from translationese. We thus expected to obtain more improvement

Table 8 Test set results obtained when using forward-translated monolingual data in addition to the parallel training data. Baseline NMT_i uses only parallel training data and baseline SMT uses monolingual data for its language model

	EN → JA		JA → EN		EN → LO		LO → EN		EN → MS		MS → EN		EN → VI		VI → EN	
	chrF		BLEU	chrF	chrF		BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF
Baselines																
NMT _i	0.212		8.6	0.372	0.339		10.5	0.374	33.3	0.605	29.4	0.554	26.8	0.468	21.9	0.484
SMT	0.243		11.7	0.415	0.355		16.1	0.436	38.3	0.649	36.5	0.619	33.8	0.527	28.3	0.548
Best baseline architecture (NMT _i) with forward translation																
18k	0.204		8.7	0.361	0.325		10.1	0.361	32.5	0.598	29.0	0.551	25.8	0.458	20.8	0.470
100k	0.212		8.2	0.363	0.335		11.5	0.375	34.0	0.610	30.1	0.564	27.3	0.473	21.9	0.481
1M	0.215		8.6	0.364	0.345		11.2	0.378	35.2	0.616	30.7	0.567	29.5	0.486	22.7	0.488
10M	0.214		8.3	0.360	–		8.0	0.340	34.7	0.620	30.5	0.562	29.0	0.484	22.7	0.486
Transformer base architecture (NMT _v) with forward translation																
18k	0.196		8.4	0.355	0.326		10.0	0.358	32.3	0.596	28.4	0.544	25.6	0.459	20.8	0.469
100k	0.208		8.7	0.365	0.334		10.8	0.374	34.3	0.613	30.0	0.559	27.8	0.474	22.1	0.481
1M	0.211		8.7	0.366	0.343		11.7	0.378	35.0	0.614	30.7	0.569	29.3	0.485	23.3	0.491
10M	0.212		8.6	0.359	–		6.4	0.308	35.3	0.621	30.2	0.560	29.2	0.485	22.9	0.487

Table 9 Test set results obtained with the multilingual NMT approach (NMT_m) along with the corresponding architectures for the eight translation directions. Two disjoint models are trained, from and towards English. Tuned parameters are: *d* model dimension, *enc* encoder layers, *dec* decoder layers, *h* heads, *norm* normalization position. NMT_i are bilingual baseline models after hyper-parameter tuning and SMT is the baseline model without monolingual data for its language model. Results in bold indicate systems significantly better than the others with $p < 0.05$

	EN → JA		JA → EN		EN → LO		LO → EN		EN → MS		MS → EN		EN → VI		VI → EN	
	chrF		BLEU	chrF	chrF		BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF
Baselines																
NMT _i	0.212		8.6	0.372	0.339		10.5	0.374	33.3	0.605	29.4	0.554	26.8	0.468	21.9	0.484
SMT	0.240		9.8	0.401	0.358		13.2	0.418	36.5	0.637	31.1	0.585	30.9	0.509	23.9	0.516
NMT _m	0.267		9.9	0.378	0.363		10.4	0.371	31.0	0.600	21.4	0.479	27.6	0.480	18.2	0.438
Best NMT _m configurations																
<i>d</i>	512		512	512	512		512	512	512	512	512	512	512	512	512	512
<i>ff</i>	512		512	512	512		512	512	512	512	512	512	512	512	512	512
<i>enc</i>	4		6	4	4		6	4	4	4	6	6	4	4	4	4
<i>dec</i>	6		2	6	6		2	6	6	6	2	2	6	6	1	1
<i>h</i>	4		1	1	1		1	1	1	1	1	1	1	1	1	1
<i>norm</i>	Pre		Pre	Pre	Pre		Pre	Pre	Pre	Pre	Pre	Pre	Pre	Pre	Post	Post

according to the automatic metrics for the $EN \rightarrow XX$ translation directions than for the $XX \rightarrow EN$ translation directions compared to the baseline NMT systems.

The results obtained for our forward translation experiments show that this approach is outperformed by SMT for all translation directions. For translation directions with translationese as target, forward translation improves over the NMT baseline, confirming the findings of previous studies. Results also show that a plateau is reached when using 1M synthetic sentence pairs and only $EN \rightarrow MS$ benefits from adding 10M pairs (Table 8).

4.3 Multilingual models

Experiments on multilingual NMT involved jointly training many-to-one and one-to-many translation models. In particular, we examined two models, one translating from English into four target languages ($EN \rightarrow XX$), and one translating from four source languages into English ($XX \rightarrow EN$). For both models, the dataset presented in Table 1 were concatenated, while the former model necessitated target language specific tokens to be pre-pended to the source sentences, as described in Sect. 2.2.

In order to compare our baseline results to the multilingual NMT approach, we conducted the same hyper-parameter search as in Sect. 4.1. Table 9 reports on the best results along with the best NMT architecture for each translation direction. These results show that the multilingual NMT approach outperforms the bilingual NMT models for 4 translation directions, namely $EN \rightarrow JA$, $JA \rightarrow EN$, $EN \rightarrow LO$, and $EN \rightarrow VI$. For 5 translation directions, SMT reaches better performances, while multilingual NMT outperforms SMT for 2 translation directions. For $JA \rightarrow EN$, SMT reaches better results compared to multilingual NMT based on the chrF metric, while both approaches are not significantly different based on BLEU.

4.4 Unsupervised multilingual models

In this section, we present the results of our UNMT experiments performed with the system described in Sect. 3.2. Previous work has shown that UNMT can reach good translation quality but experiments were limited to high-resource language pairs. We intended to investigate the usefulness of UNMT for our extremely low-resource language pairs, including a truly low-resource language, Lao, for which we only had less than 1M lines of monolingual data. To the best of our knowledge, this is the first time that UNMT experiments and results for such small amount of monolingual data are reported. Results are presented in Table 10.

We observed that our UNMT model (noted NMT_u) is unable to produce translations for the $EN-JA$ language pair in both translation directions as exhibited by BLEU and chrF scores close to 0. On the other hand, NMT_u performs slightly better than NMT_l for $LO \rightarrow EN$ (+0.2 BLEU, +0.004 chrF) and $MS \rightarrow EN$ (+2.0 BLEU, +0.011 chrF). However, tagged backward translation virtually requires less monolingual data than NMT_u , but exploits also the original parallel data, leading to the best system by a large margin of more than 10 BLEU points for all language pairs except $EN-LO$.

Table 10 Test set results obtained with UNMT (NMT_u) in comparison with the best $NMT \rightarrow$ systems, trained without using monolingual data, and systems trained on 10M (or 737k for $EN \rightarrow LO$) tagged backward translations with the Transformer *base* architecture (T-BT). SMT uses monolingual data for its language model

	EN $\rightarrow$ JA		JA $\rightarrow$ EN		EN $\rightarrow$ LO		LO $\rightarrow$ EN		EN $\rightarrow$ MS		MS $\rightarrow$ EN		EN $\rightarrow$ VI		VI $\rightarrow$ EN	
	chrF		BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF
Baselines																
NMT_t	0.212		8.6	0.372	0.339		10.5	0.374	33.3	0.605	29.4	0.554	26.8	0.468	21.9	0.484
SMT	0.243		11.7	0.415	0.355		16.1	0.436	38.3	0.649	36.5	0.619	33.8	0.527	28.3	0.548
Transformer base architecture (NMT_v) with tagged backward translation																
T-BT	0.253		17.5	0.471	0.382		18.4	0.459	40.5	0.662	40.0	0.639	35.2	0.535	32.0	0.568
Unsupervised NMT																
NMT_u	0.017		0.0	0.036	0.333		10.7	0.378	27.8	0.563	31.4	0.565	22.0	0.429	21.9	0.498

5 Analysis and discussion

This section presents the analysis and discussion relative to the experiments and results obtained in Sect. 4, focusing on four aspects.

1. The position of the layer normalization component and its impact on the convergence of NMT models as well as on the final results according to automatic metrics.
2. The integration of monolingual data produced by baseline NMT models with tuned hyper-parameters through four different approaches.
3. The combination of the eight translation directions in a multilingual NMT model.
4. The training of unsupervised NMT by using monolingual data only.

5.1 Layer normalization position

Layer normalization (Ba et al. 2016) is a crucial component of the Transformer architecture, since it allows for stable training and fast convergence. Some recent studies have shown that the position of the layer normalization mechanism has an impact on the ability to train deep Transformer models in high-resource settings or out-of-the-box Transformer configurations in low-resource settings. For the latter, Nguyen and Salazar (2019) show that when using a Transformer *base* configuration, one should always put layer normalization prior to the non-linear layers within a Transformer block (i.e., pre-norm).

Our experiments on eight translation directions in extremely low-resource settings validate these findings. More precisely, all our baseline NMT systems using the Transformer *base* configuration with post-non-linearity layer normalization (i.e. post-norm) do not converge. Successfully trained baseline models using post-norm require fewer number of encoder layers, while pre-norm allows for deeper architecture even with extremely low-resource language pairs.

An interesting observation made on the baseline results is that the decoder part of the Transformer does not suffer from depth-related issue. Additionally, during hyper-parameter tuning, results show that there are larger gaps in terms of chrF scores between configurations using post-norm than the ones using pre-norm. As a recommendation, we encourage the NMT practitioners to do hyper-parameter search including at least two layer normalization positions, pre- and post-norm, in order to reach the best performance in low-resource settings. Another recommendation, in order to save computing time during hyper-parameter search, is to limit the encoder depth to a maximum of 4 layers when using post-norm, as our experiments show that 6-layer encoders and post-norm do not converge for all language pairs and translation directions of the training corpora presented in this paper.

Fig. 1 Test set results obtained by the best tuned baseline (NMT_t) and Transformer *base* (NMT_v) with various amount of monolingual data for back-translation (*bt*), tagged back-translation (*t-bt*) and forward translation (*ft*)

Table 11 Test set results obtained with the best NMT systems using backward translations (BT), tagged backward translations (T-BT), forward translations (FT), and noised backward translations (N-BT), regardless of the hyper-parameters and quantity of monolingual data used. Results in bold indicate systems significantly better than the others with $p < 0.05$

	EN $\rightarrow$ JA		JA $\rightarrow$ EN		EN $\rightarrow$ LO		LO $\rightarrow$ EN		EN $\rightarrow$ MS		MS $\rightarrow$ EN		EN $\rightarrow$ VI		VI $\rightarrow$ EN	
	chrF		BLEU	chrF	chrF		BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF
BT	0.240		14.5	0.444	0.364		17.4	0.455	38.1	0.646	39.7	0.638	32.9	0.519	30.7	0.558
T-BT	0.253		17.5	0.471	0.382		18.4	0.459	40.5	0.662	40.1	0.639	35.2	0.535	32.0	0.568
N-BT	0.234		14.6	0.445	0.367		17.4	0.448	37.7	0.643	39.5	0.633	33.4	0.522	30.5	0.556
FT	0.215		8.7	0.366	0.345		11.7	0.378	35.3	0.621	30.7	0.569	29.5	0.486	22.9	0.491

5.2 Monolingual data

Producing synthetic parallel data from monolingual corpora has been shown to improve NMT performance. In our work, we investigate four methods to generate such parallel data, including backward and forward translations. For the former method, three variants of the commonly used backward translation are explored, with or without source-side tags indicating the origin of the data, and with or without noise introduced in the source sentences.

When using backward translations without tags and up to 10M monolingual sentences, the Transformer *base* architecture outperforms the best baseline architecture obtained through hyper-parameter search. However, the synthetic data itself was produced by the best baseline NMT system. Our preliminary experiments using the Transformer *base* to produce the backward translations did unsurprisingly lead to lower translation quality, due to the low quality of the synthetic data. Nevertheless, using more than 100k backward translations produced by the best baseline architecture allows to switch to the Transformer *base*. As indicated by Fig. 1, when using backward translated synthetic data (noted *bt*), Transformer *base* (NMT_v) outperforms the tuned baseline (NMT_t) following the same trend for the eight translation directions.

Based on the empirical results obtained in Table 6, tagged backward translation is the best performing approach among all other approaches evaluated in this work, according to automatic metrics, and regardless of the quantity of monolingual data used, as summarized in Table 11. Noised backward translations and forward translations both underperform tagged backward translation. Despite our extremely low-resource setting, we could confirm the findings of Caswell et al. (2019) that pre-pending a tag to each backward-translated sentence is very effective in improving BLEU scores, irrespective of whether the translation direction is for original or translationese texts. However, while Caswell et al. (2019) reports that tagged backward translation is as effective as introducing synthetic noise in a high-resource settings, we observe a different tendency in our experiments. Using the noised backward translation approach leads to lower scores than tagged backward translation and does not improve over the use of backward translations without noise.

Since we only have very small original parallel data, we assume that degrading the quality of the backward translations is too detrimental to train an NMT system that does not have enough instance of well-formed source sentences to learn the characteristics of the source language. Edunov et al. (2018) also report on a similar observation in a low-resource configuration using only 80k training parallel sentences. As for the use of forward translations, this approach outperforms the backward translation approach only when using a small quantity of monolingual data. As discussed by Bogoychev and Sennrich (2019), training NMT with synthetic data on the target side can be misleading for the decoder especially when the forward translations are of a very poor quality such as translations generated by a low-resource NMT system.

5.3 Multilingualism

Comparing the BLEU scores of bilingual and multilingual models for the EN $\rightarrow$ JA, EN $\rightarrow$ LO and EN $\rightarrow$ VI translation directions, multilingual models outperform bilingual ones. On the other hand, for the other translation directions, the performance of multilingual models are comparable or lower than bilingual ones. When comparing SMT to multilingual NMT models, well tuned multilingual models outperform SMT for the EN $\rightarrow$ JA and EN $\rightarrow$ LO translation directions, while SMT is as good or better for the remaining six translation directions. In contrast, well tuned NMT models are unable to outperform SMT regardless of the translation direction.

Our observations are in line with previous works such as Firat et al. (2016), Johnson et al. (2017) which showed that multilingual models outperform bilingual ones for low-resource language pairs. However, these works did not focus on extremely low-resource setting and multi-parallel corpora for training multilingual models. Although multilingualism is known to improve performance for low-resource languages, we observe drops in performance for some of the language pairs involved. The work on multi-stage fine-tuning (Dabre et al. 2019), which uses N-way parallel corpora similar to the ones used in our work, supports our observations regarding drops in performance. However, based on the characteristics of our parallel corpora, the multilingual models trained and evaluated in our work do not benefit from additional knowledge by increasing the number of translation directions, because the translation content for all language pairs is the same. Although our multilingual models do not always outperform bilingual NMT or SMT models, we observe that they are useful for difficult translation directions where BLEU scores are below 10pts.

With regards to the tuned hyper-parameters, we noticed slight differences between multilingual and bilingual models. By comparing the best configurations in Tables 9 and 4, we observe that the best multilingual models are mostly those that use pre-normalization of layers (7 translation directions out of 8). In contrast, there is no such tendency for bilingual models, where 3 configurations out of 8 reach the best performance using pre-normalization.

Another observation is related to the number of decoder layers, where shallower decoders are better for translating into English whereas deeper ones are better for the opposite direction. This tendency is different from the one that bilingual models exhibit where deeper decoder layers are almost always preferred (6 layers in 7 configurations out of 8). We assume that the many-to-English models require a shallower decoder architecture because of the repetitions on the target side of the training and validation data. However, a deeper analysis is required, involving other language pairs and translation directions, to validate this hypothesis. To the best of our knowledge, in the context of multilingual NMT models for extremely low-resource settings, the study of vast hyperparameter tuning, optimal configurations and performance comparisons do not exist.

5.4 Unsupervised NMT

Our main observation from the UNMT experiments is that it only performs similarly or slightly better, respectively for EN-LO and MS $\rightarrow$ EN, than our best supervised NMT systems with tuned hyper-parameters, while it is significantly worse for all the remaining translation directions. These results are particularly surprising for EN-LO for which we have only very few data to pre-train an XLM model and UNMT, meaning that we do not necessarily need a large quantity of monolingual data for each language in a multilingual scenario for UNMT. Nonetheless, the gap between supervised MT and UNMT is even more significant if we take as baseline SMT systems with a difference of more than 5.0 BLEU for all translation directions. Training an MT system with only 18k parallel data is thus preferable to using only monolingual data for these extremely low-resource configurations. Furthermore, using only parallel data for supervised NMT is unrealistic since we have also access at least to the same monolingual data used by UNMT. The gap is then even greater when exploiting monolingual data as tagged backward translation for training supervised NMT.

Our results are in line with the few previous work on unsupervised MT for distant low-resource language pairs (Marie et al. 2019) but to the best of our knowledge this is the first time that results for such language pairs are reported using an UNMT system initialized with pre-trained cross-lingual language model (XLM). We assume that the poor performances reached by our UNMT models are mainly due to the significant lexical and syntactic differences between English and the other languages, making the training of a single multilingual embedding space for these languages a very hard task (Søgaard et al. 2018). This is well exemplified by the EN-JA language pair, that involves two languages with a completely different writing systems, for which BLEU and chrF scores are close to 0. The performance of UNMT is thus far from the performance of supervised MT. Further research is necessary in UNMT for distant low-resource language pairs in order to improve it and rivaling the performance of supervised NMT, as observed for more similar language pairs such as French–English and English–German (Artetxe et al. 2018; Lample et al. 2018; Artetxe et al. 2018; Lample et al. 2018).

6 Conclusion

This paper presented a study on extremely low-resource language pairs with the Transformer NMT architecture involving Asian languages and eight translation directions. After conducting an exhaustive hyper-parameter search focusing on the specificity of the Transformer to define our baseline systems, we trained and evaluated translation models making use of various amount of synthetic data. Four different approaches were employed to generate synthetic data, including backward translations, with or without specific tags and added noise, and forward translations, which were then contrasted with an unsupervised NMT approach built only using monolingual data. Finally, based on the characteristics of the parallel data used in our experiments, we jointly trained multilingual NMT systems from and towards English.

The main objectives of the work presented in this paper is to deliver a *recipe* allowing MT practitioners to train NMT systems in extremely low-resource scenarios. Based on the empirical evidences validated by eight translation directions, we make the following recommendations.

- First, an exhaustive hyper-parameter search, including the position of layer normalization within the Transformer block, is crucial for both a strong baseline and producing synthetic data of sufficient quality.
- Second, a clear preference for backward compared to forward translations for the synthetic data generation approach.
- Third, generating enough backward translations to benefit from the large amount of parameters available in the commonly used Transformer architecture in terms of number and dimensionality of layers.
- Fourth, adding a tag on the source side of backward translations to indicate its origin, which leads to the higher performance than not adding tag or introducing noise.

As future work, we plan to enlarge the search space for hyper-parameter tuning, including more general parameters which are not specific to the Transformer architecture, such as various dropout rates, vocabulary sizes and learning rates. Additionally, we want to increase the number of learnable parameters in the Transformer architecture to avoid reaching a plateau when the amount of synthetic training data increases. Finally, we will explore other multilingual NMT approaches in order to improve the results obtained in this work and to make use of tagged backward translation as we did for the bilingual models.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s10590-020-09258-6>.

Acknowledgements A part of this work was conducted under the program “Research and Development of Enhanced Multilingual and Multipurpose Speech Translation System” of the Ministry of Internal Affairs and Communications (MIC), Japan. Benjamin Marie was partly supported by JSPS KAKENHI Grant Number 20K19879 and the tenure-track researcher start-up fund in NICT. Atsushi Fujita and Masao Utiyama were partly supported by JSPS KAKENHI Grant Number 19H05660.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

Aharoni R, Johnson M, Firat O (2019) Massively multilingual neural machine translation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational

- Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp 3874–3884. Association for Computational Linguistics, Minneapolis, USA. <https://doi.org/10.18653/v1/N19-1388>. <https://aclweb.org/anthology/N19-1388>
- Artetxe M, Labaka G, Agirre E (2018) Unsupervised statistical machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp 3632–3642. Association for Computational Linguistics, Brussels, Belgium. <https://doi.org/10.18653/v1/D18-1399>. <https://aclweb.org/anthology/D18-1399>
- Artetxe M, Labaka G, Agirre E, Cho K (2018) Unsupervised neural machine translation. In: Proceedings of the 6th international conference on learning representations. Vancouver, Canada. <https://openreview.net/forum?id=Sy2ogeBAW>
- Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. [arXiv:1607.06450](https://arxiv.org/abs/1607.06450)
- Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: Proceedings of the 3rd international conference on learning representations. San Diego, USA. [arxiv:1409.0473](https://arxiv.org/abs/1409.0473)
- Barraut L, Bojar O, Costa-jussà MR, Federmann C, Fishel M, Graham Y, Haddow B, Huck M, Koehn P, Malmasi S, Monz C, Müller M, Pal S, Post M, Zampieri M (2019) Findings of the 2019 conference on machine translation (WMT19). In: Proceedings of the fourth conference on machine translation (Volume 2: Shared Task Papers, Day 1), pp 1–61. Association for Computational Linguistics, Florence, Italy. <https://doi.org/10.18653/v1/W19-5301>. <https://aclweb.org/anthology/W19-5301>
- Bogoychev N, Sennrich R (2019) Domain, translationese and noise in synthetic data for neural machine translation. [arXiv:1911.03362](https://arxiv.org/abs/1911.03362)
- Bojanowski P, Grave E, Joulin A, Mikolov T (2016) Enriching word vectors with subword information. [arXiv:1607.04606](https://arxiv.org/abs/1607.04606)
- Bojar O, Chatterjee R, Federmann C, Graham Y, Haddow B, Huck M, Jimeno Yepes A, Koehn P, Logacheva V, Monz C, Negri M, Neveol A, Neves M, Popel M, Post M, Rubino R, Scarton C, Specia L, Turchi M, Verspoor K, Zampieri M (2016) Findings of the 2016 conference on machine translation. In: Proceedings of the first conference on machine translation, pp 131–198. Association for Computational Linguistics, Berlin, Germany. <https://doi.org/10.18653/v1/W16-2301>. <https://aclweb.org/anthology/W16-2301>
- Brown PF, Lai JC, Mercer RL (1991) Aligning sentences in parallel corpora. In: Proceedings of the 29th annual meeting on association for computational linguistics, pp 169–176. Association for Computational Linguistics, Berkeley, USA. <https://doi.org/10.3115/981344.981366>. <https://aclweb.org/anthology/P91-1022>
- Burlot F, Yvon F (2018) Using monolingual data in neural machine translation: a systematic study. In: Proceedings of the third conference on machine translation: research papers, pp 144–155. Association for Computational Linguistics, Brussels, Belgium. <https://doi.org/10.18653/v1/W18-6315>. <https://aclweb.org/anthology/W18-6315>
- Caswell I, Chelba C, Grangier D (2019) Tagged back-translation. In: Proceedings of the fourth conference on machine translation (volume 1: research papers), pp 53–63. association for computational linguistics, Florence, Italy. <https://doi.org/10.18653/v1/W19-5206>. <https://aclweb.org/anthology/W19-5206.pdf>
- Cettolo M, Jan N, Sebastian S, Bentivogli L, Cattoni R, Federico M (2016) The IWSLT 2016 evaluation campaign. In: Proceedings of the 13th international workshop on spoken language translation. Seattle, USA. https://workshop2016.iwslt.org/downloads/IWSLT_2016_evaluation_overview.pdf
- Chen PJ, Shen J, Le M, Chaudhary V, El-Kishky A, Wenzek G, Ott M, Ranzato M (2019) Facebook AI's WAT19 Myanmar-English Translation Task submission. In: Proceedings of the 6th Workshop on Asian Translation, pp 112–122. Association for Computational Linguistics, Hong Kong, China. <https://doi.org/10.18653/v1/D19-5213>. <https://aclweb.org/anthology/D19-5213>
- Cherry C, Foster G (2012) Batch tuning strategies for statistical machine translation. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 427–436. Association for Computational Linguistics, Montréal, Canada. <https://aclweb.org/anthology/N12-1047>
- Cho K, van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: Encoder–decoder approaches. In: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pp. 103–111. Association for Computational Linguistics, Doha, Qatar. <https://doi.org/10.3115/v1/W14-4012>. <https://aclweb.org/anthology/W14-4012/>

- Dabre R, Fujita A, Chu C (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), pp. 1410–1416. Association for Computational Linguistics, Hong Kong, China. <https://doi.org/10.18653/v1/D19-1146>. <https://aclweb.org/anthology/D19-1146>
- Eduvov S, Ott M, Auli M, Grangier D (2018) Understanding back-translation at scale. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 489–500. Association for Computational Linguistics, Brussels, Belgium. <https://doi.org/10.18653/v1/D18-1045>. <https://aclweb.org/anthology/D18-1045>
- Elsken T, Metzen JH, Hutter F (2018) Neural architecture search: a survey. [arXiv:1808.05377](https://arxiv.org/abs/1808.05377)
- Firat O, Cho K, Bengio Y (2016) Multi-way, multilingual neural machine translation with a shared attention mechanism. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 866–875. Association for Computational Linguistics, San Diego, USA. <https://doi.org/10.18653/v1/N16-1101>. <https://aclweb.org/anthology/N16-1101>
- Firat O, Sankaran B, Al-Onaizan Y, Yarman Vural FT, Cho K (2016) Zero-resource translation with multi-lingual neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 268–277. Association for Computational Linguistics, Austin, USA. <https://doi.org/10.18653/v1/D16-1026>. <https://aclweb.org/anthology/D16-1026>
- Forcada ML, Ñeco RP (1997) Recursive hetero-associative memories for translation. In: Mira J, Moreno-Díaz R, Cabestany J (eds) Biological and artificial computation: from neuroscience to technology. Springer, Berlin. <https://doi.org/10.1007/BFb0032504>
- Gellerstam M (1986) Translationese in Swedish novels translated from English. *Transl Stud Scand* 1:88–95
- Gulcehre C, Firat O, Xu K, Cho K, Bengio Y (2017) On integrating a language model into neural machine translation. *Comput Speech Lang* 45:137–148. <https://doi.org/10.1016/j.csl.2017.01.014>
- Heafield K, Pouzyrevsky I, Clark JH, Koehn P (2013) Scalable modified Kneser-Ney language model estimation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pp. 690–696. Association for Computational Linguistics, Sofia, Bulgaria. <https://aclweb.org/anthology/P13-2121/>
- Imamura K, Sumita E (2018) NICT self-training approach to neural machine translation at NMT-2018. In: Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pp. 110–115. Association for Computational Linguistics, Melbourne, Australia. <https://doi.org/10.18653/v1/W18-2713>. <https://aclweb.org/anthology/W18-2713>
- Johnson M, Schuster M, Le QV, Krikun M, Wu Y, Chen Z, Thorat N, Viégas F, Wattenberg M, Corrado G, Hughes M, Dean J (2017) Google’s multilingual neural machine translation system: enabling zero-shot translation. *Trans Assoc Comput Linguist* 5:339–351. https://doi.org/10.1162/tacl_a_00065
- Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
- Koehn P (2004) Statistical significance tests for machine translation evaluation. In: Proceedings of the 2004 conference on empirical methods in natural language processing, pp. 388–395
- Koehn P, Huang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 177–180. Association for Computational Linguistics, Prague, Czech Republic. <https://aclweb.org/anthology/P07-2045>
- Koehn P, Knowles R (2017) Six challenges for neural machine translation. In: Proceedings of the First Workshop on Neural Machine Translation, pp. 28–39. Association for Computational Linguistics, Vancouver, Canada. <https://doi.org/10.18653/v1/W17-3204>. <https://aclweb.org/anthology/W17-3204>
- Kudo T, Richardson J (2018) Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 66–71. Association for Computational Linguistics, Brussels, Belgium. <https://doi.org/10.18653/v1/D18-2012>. <https://aclweb.org/anthology/D18-2012>
- Lambert P, Schwenk H, Servan C, Abdul-Rauf S (2011) Investigations on translation model adaptation using monolingual data. In: Proceedings of the Sixth Workshop on Statistical Machine Translation,

- pp. 284–293. Association for Computational Linguistics, Edinburgh, Scotland. <https://aclweb.org/anthology/W11-2132>
- Lample G, Conneau A (2019) Cross-lingual language model pretraining. In: Proceedings of Advances in Neural Information Processing Systems 32, pp. 7057–7067. Curran Associates, Inc., Vancouver, Canada. <https://papers.nips.cc/paper/8928-cross-lingual-language-model-pretraining>
- Lample G, Conneau A, Denoyer L, Ranzato M (2018) Unsupervised machine translation using monolingual corpora only. In: Proceedings of the 6th International Conference on Learning Representations. Vancouver, Canada. <https://openreview.net/forum?id=rkYTTf-AZ>
- Lample G, Ott M, Conneau A, Denoyer L, Ranzato M (2018) Phrase-based & neural unsupervised machine translation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 5039–5049. Association for Computational Linguistics, Brussels, Belgium. <https://doi.org/10.18653/v1/D18-1549>. <https://aclweb.org/anthology/D18-1549>
- Marie B, Dabre R, Fujita A (2019) NICT's machine translation systems for the WMT19 similar language translation task. In: Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2), pp. 208–212. Association for Computational Linguistics, Florence, Italy. <https://doi.org/10.18653/v1/W19-5428>. <https://aclweb.org/anthology/W19-5428>
- Marie B, Sun H, Wang R, Chen K, Fujita A, Utiyama M, Sumita E (2019) NICT's unsupervised neural and statistical machine translation systems for the WMT19 news translation task. In: Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1), pp. 294–301. Association for Computational Linguistics, Florence, Italy. <https://doi.org/10.18653/v1/W19-5330>. <https://aclweb.org/anthology/W19-5330>
- Michel P, Levy O, Neubig G (2019) Are sixteen heads really better than one? In: Proceedings of Advances in Neural Information Processing Systems 32, pp. 14014–14024. Curran Associates, Inc., Vancouver, Canada. <https://papers.nips.cc/paper/9551-are-sixteen-heads-really-better-than-one>
- Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, pp. 807–814. Madison, USA. <https://dl.acm.org/doi/10.5555/3104322.3104425>
- Nguyen TQ, Salazar J (2019) Transformers without tears: improving the normalization of self-attention. [arXiv:1910.05895](https://arxiv.org/abs/1910.05895)
- Och FJ, Ney H (2000) Improved statistical alignment models. In: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, pp. 440–447. Association for Computational Linguistics, Hong Kong, China. <https://doi.org/10.3115/1075218.1075274>. <https://aclweb.org/anthology/P00-1056/>
- Ott M, Edunov S, Baevski A, Fan A, Gross S, Ng N, Grangier D, Auli M (2019) fairseq: A fast, extensible toolkit for sequence modeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pp. 48–53. Association for Computational Linguistics, Minneapolis, USA. <https://doi.org/10.18653/v1/N19-4009>. <https://aclweb.org/anthology/N19-4009/>
- Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318. Association for Computational Linguistics, Philadelphia, USA. <https://doi.org/10.3115/1073083.1073135>. <https://aclweb.org/anthology/P02-1040/>
- Popović M (2015) chrF: character n-gram F-score for automatic MT evaluation. In: Proceedings of the Tenth Workshop on Statistical Machine Translation, pp. 392–395. Association for Computational Linguistics, Lisbon, Portugal. <https://doi.org/10.18653/v1/W15-3049>. <https://aclweb.org/anthology/W15-3049/>
- Post M (2018) A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 186–191. Association for Computational Linguistics, Brussels, Belgium. <https://doi.org/10.18653/v1/W18-6319>. <https://aclweb.org/anthology/W18-6319/>
- Riza H, Purwoadi M, Uliniansyah T, Ti AA, Aljunied SM, Mai LC, Thang VT, Thai NP, Chea V, Sam S, et al (2016) Introduction of the Asian Language Treebank. In: 2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA), pp. 1–6. Institute of Electrical and Electronics Engineers, Bali, Indonesia. <https://doi.org/10.1109/ICSDA.2016.7918974>. <https://ieeexplore.ieee.org/document/7918974>

- Schwenk H (2008) Investigations on large-scale lightly-supervised training for statistical machine translation. In: Proceedings of the International Workshop on Spoken Language Translation, pp. 182–189. Honolulu, USA. https://www.isca-speech.org/archive/iwslt_08/papers/slt8_182.pdf
- Schwenk H, Chaudhary V, Sun S, Gong H, Guzmán F (2019) WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia
- Sennrich R, Haddow B, Birch A (2016a) Improving neural machine translation models with monolingual data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 86–96. Association for Computational Linguistics, Berlin, Germany. <https://doi.org/10.18653/v1/P16-1009>. <https://aclweb.org/anthology/P16-1009>
- Sennrich R, Haddow B, Birch A (2016b) Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1715–1725. Association for Computational Linguistics, Berlin, Germany. <https://doi.org/10.18653/v1/P16-1162>. <https://aclweb.org/anthology/P16-1162>
- Sennrich R, Zhang B (2019) Revisiting low-resource neural machine translation: A case study. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 211–221. Association for Computational Linguistics, Florence, Italy. <https://doi.org/10.18653/v1/P19-1021>. <https://aclweb.org/anthology/P19-1021>
- Søgaard A, Ruder S, Vulić I (2018) On the limitations of unsupervised bilingual dictionary induction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 778–788. Association for Computational Linguistics, Melbourne, Australia. <https://doi.org/10.18653/v1/P18-1072>. <https://aclweb.org/anthology/P18-1072>
- Stahlberg F, Cross J, Stoyanov V (2018) Simple fusion: Return of the language model. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 204–211. Association for Computational Linguistics, Brussels, Belgium. <https://doi.org/10.18653/v1/W18-6321>. <https://aclweb.org/anthology/W18-6321>
- Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Proceedings of the 27th Neural Information Processing Systems Conference (NIPS), pp. 3104–3112. Curran Associates, Inc., Montréal, Canada. <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>
- Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Proceedings of the 30th Neural Information Processing Systems Conference (NIPS), pp. 5998–6008. Curran Associates, Inc., Long Beach, USA. <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>
- Voita E, Talbot D, Moiseev F, Sennrich R, Titov I (2019) Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5797–5808. Association for Computational Linguistics, Florence, Italy. <https://doi.org/10.18653/v1/P19-1580>. <https://aclweb.org/anthology/P19-1580/>
- Wang Q, Li B, Xiao T, Zhu J, Li C, Wong DF, Chao LS (2019) Learning deep transformer models for machine translation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1810–1822. Association for Computational Linguistics, Florence, Italy. <https://doi.org/10.18653/v1/P19-1176>. <https://aclweb.org/anthology/P19-1176>
- Zens R, Och FJ, Ney H (2002) Phrase-based statistical machine translation. In: Annual Conference on Artificial Intelligence, pp. 18–32. Springer

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.