Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism

Orhan Firat
Middle East Technical University
orhan.firat@ceng.metu.edu.tr

Kyunghyun Cho
New York University

Yoshua Bengio
University of Montreal
CIFAR Senior Fellow

Abstract

We propose multi-way, multilingual neural machine translation. The proposed approach enables a single neural translation model to translate between multiple languages, with a number of parameters that grows only linearly with the number of languages. This is made possible by having a single attention mechanism that is shared across all language pairs. We train the proposed multi-way, multilingual model on ten language pairs from WMT'15 simultaneously and observe clear performance improvements over models trained on only one language pair. In particular, we observe that the proposed model significantly improves the translation quality of low-resource language pairs.

1 Introduction

Neural Machine Translation It has been shown that a deep (recurrent) neural network can successfully learn a complex mapping between variable-length input and output sequences on its own. Some of the earlier successes in this task have, for instance, been handwriting recognition (Bottou et al., 1997; Graves et al., 2009) and speech recognition (Graves et al., 2006; Chorowski et al., 2015). More recently, a general framework of encoder-decoder networks has been found to be effective at learning this kind of sequence-to-sequence mapping by using two recurrent neural networks (Cho et al., 2014b; Sutskever et al., 2014).

A basic encoder-decoder network consists of two recurrent networks. The first network, called an encoder, maps an input sequence of variable length into a point in a continuous vector space, resulting in a fixed-dimensional context vector. The other recurrent neural network, called a decoder, then generates a target sequence again of variable length starting from the context vector. This approach however has been found to be inefficient in (Cho et al., 2014a) when handling long sentences, due to the difficulty in learning a complex mapping between an arbitrary long sentence and a single fixed-dimensional vector.

In (Bahdanau et al., 2014), a remedy to this issue was proposed by incorporating an *attention mechanism* to the basic encoder-decoder network. The attention mechanism in the encoder-decoder network frees the network from having to map a sequence of arbitrary length to a single, fixed-dimensional vector. Since this attention mechanism was introduced to the encoder-decoder network for machine translation, neural machine translation, which is purely based on neural networks to perform full end-to-end translation, has become competitive with the existing phrase-based statistical machine translation in many language pairs (Jean et al., 2015; Gulcehre et al., 2015; Luong et al., 2015b).

Multilingual Neural Machine Translation Existing machine translation systems, mostly based on a phrase-based system or its variants, work by directly mapping a symbol or a subsequence of symbols in a source language to its corresponding symbol or subsequence in a target language. This kind of mapping is strictly specific to a given language *pair*, and it is not trivial to extend this mapping to work on multiple pairs of languages.

A system based on neural machine translation, on the other hand, can be decomposed into two mod-

Proceedings of NAACL-HLT 2016, pages 866–875,
San Diego, California, June 12-17, 2016.

ules. The encoder maps a source sentence into a continuous representation, either a fixed-dimensional vector in the case of the basic encoder-decoder network or a set of vectors in the case of attention-based encoder-decoder network. The decoder then generates a target translation based on this source representation. This makes it possible conceptually to build a system that maps a source sentence in any language to a common continuous representation space and decodes the representation into any of the target languages, allowing us to make a *multilingual machine translation* system.

This possibility is straightforward to implement and has been validated in the case of basic encoder-decoder networks (Luong et al., 2015a). It is however not so, in the case of the attention-based encoder-decoder network, as the attention mechanism, or originally called the alignment function in (Bahdanau et al., 2014), is conceptually language pair-specific. In (Dong et al., 2015), the authors cleverly avoided this issue of language pair-specific attention mechanism by considering only a one-to-many translation, where each target language decoder embedded its own attention mechanism. Also, we notice that both of these works have only evaluated their models on relatively small-scale tasks, making it difficult to assess whether multilingual neural machine translation can scale beyond low-resource language translation.

Multi-Way, Multilingual Neural Machine Translation In this paper, we first step back from the currently available multilingual neural translation systems proposed in (Luong et al., 2015a; Dong et al., 2015) and ask the question of whether the attention mechanism can be shared across multiple language pairs. As an answer to this question, we propose an attention-based encoder-decoder network that admits a shared attention mechanism with multiple encoders and decoders. We use this model for all the experiments, which suggests that it is indeed possible to share an attention mechanism across multiple language pairs.

The next question we ask is the following: in which scenario would the proposed multi-way, multilingual neural translation have an advantage over the existing, single-pair model? Specifically, we consider a case of the translation between a low-resource language pair. The experiments show that the proposed multi-way, multilingual model generalizes better than the single-pair translation model, when the amount of available parallel corpus is small. Furthermore, we validate that this is not only due to the increased amount of target-side, monolingual corpus.

Finally, we train a single model with the proposed architecture on all the language pairs from the WMT'15; English, French, Czech, German, Russian and Finnish. The experiments show that it is indeed possible to train a single attention-based network to perform multi-way translation.

2 Background: Attention-based Neural Machine Translation

The attention-based neural machine translation was proposed in (Bahdanau et al., 2014). It was motivated from the observation in (Cho et al., 2014a) that a basic encoder-decoder translation model from (Cho et al., 2014b; Sutskever et al., 2014) suffers from translating a long source sentence efficiently. This is largely due to the fact that the encoder of this basic approach needs to compress a whole source sentence into a single vector. Here we describe the attention-based neural machine translation.

Neural machine translation aims at building a single neural network that takes as input a source sequence $X = (x_1, \ldots, x_{T_x})$ and generates a corresponding translation $Y = \left(y_1, \ldots, y_{T_y}\right)$. Each symbol in both source and target sentences, x_t or y_t, is an integer index of the symbol in a vocabulary.

The encoder of the attention-based model encodes a source sentence into a set of context vectors $C = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{T_x}\}$, whose size varies w.r.t. the length of the source sentence. This context set is constructed by a bidirectional recurrent neural network (RNN) which consists of a forward RNN and reverse RNN. The forward RNN reads the source sentence from the first token until the last one, resulting in the forward context vectors $\left\{\overrightarrow{\mathbf{h}}_1, \ldots, \overrightarrow{\mathbf{h}}_{T_x}\right\}$, where

$$\overrightarrow{\mathbf{h}}_t = \overrightarrow{\Psi}_{\text{enc}}\left(\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{E}_x\left[x_t\right]\right),$$

and $\mathbf{E}_x \in \mathbb{R}^{|V_x| \times d}$ is an embedding matrix containing row vectors of the source symbols. The

reverse RNN in an opposite direction, resulting in $\left\{\overleftarrow{\mathbf{h}}_1, \ldots, \overleftarrow{\mathbf{h}}_{T_x}\right\}$, where

$$\overleftarrow{\mathbf{h}}_t = \overleftarrow{\Psi}_{\text{enc}}\left(\overleftarrow{\mathbf{h}}_{t+1}, \mathbf{E}_x\left[x_t\right]\right).$$

$\overrightarrow{\Psi}_{\text{enc}}$ and $\overleftarrow{\Psi}_{\text{enc}}$ are recurrent activation functions such as long short-term memory units (LSTM, (Hochreiter and Schmidhuber, 1997)) or gated recurrent units (GRU, (Cho et al., 2014b)). At each position in the source sentence, the forward and reverse context vectors are concatenated to form a full context vector, i.e.,

$$\mathbf{h}_t = \left[\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t\right]. \quad (1)$$

The decoder, which is implemented as an RNN as well, generates one symbol at a time, the translation of the source sentence, based on the context set returned by the encoder. At each time step t in the decoder, a time-dependent context vector $\mathbf{c}_t$ is computed based on the previous hidden state of the decoder $\mathbf{z}_{t-1}$, the previously decoded symbol $\tilde{y}_{t-1}$ and the whole context set C.

This starts by computing the relevance score of each context vector as

$$e_{t,i} = f_{\text{score}}(\mathbf{h}_i, \mathbf{z}_{t-1}, \mathbf{E}_y\left[\tilde{y}_{t-1}\right]), \quad (2)$$

for all $i = 1, \ldots, T_x$. f_{score} can be implemented in various ways (Luong et al., 2015b), but in this work, we use a simple single-layer feedforward network. This relevance score measures how relevant the i-th context vector of the source sentence is in deciding the next symbol in the translation. These relevance scores are further normalized:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{T_x}\exp(e_{t,j})}, \quad (3)$$

and we call $\alpha_{t,i}$ the attention weight.

The time-dependent context vector $\mathbf{c}_t$ is then the weighted sum of the context vectors with their weights being the attention weights from above:

$$\mathbf{c}_t = \sum_{i=1}^{T_x}\alpha_{t,i}\mathbf{h}_i. \quad (4)$$

With this time-dependent context vector $\mathbf{c}_t$, the previous hidden state $\mathbf{z}_{t-1}$ and the previously decoded symbol $\tilde{y}_{t-1}$, the decoder's hidden state is updated by

$$\mathbf{z}_t = \Psi_{\text{dec}}\left(\mathbf{z}_{t-1}, \mathbf{E}_y\left[\tilde{y}_{t-1}\right], \mathbf{c}_t\right), \quad (5)$$

where Ψ_{dec} is a recurrent activation function.

The initial hidden state $\mathbf{z}_0$ of the decoder is initialized based on the last hidden state of the reverse RNN:

$$\mathbf{z}_0 = f_{\text{init}}\left(\overleftarrow{\mathbf{h}}_{T_x}\right), \quad (6)$$

where f_{init} is a feedforward network with one or two hidden layers.

The probability distribution for the next target symbol is computed by

$$p(y_t = k|\tilde{y}_{<t}, X) \propto e^{g_k(\mathbf{z}_t, \mathbf{c}_t, \mathbf{E}[\tilde{y}_{t-1}])}, \quad (7)$$

where g_k is a parametric function that returns the unnormalized probability for the next target symbol being k.

Training this attention-based model is done by maximizing the conditional log-likelihood

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_y}\log p(y_t = y_t^{(n)}|y_{<t}^{(n)}, X^{(n)}), \quad (8)$$

where the log probability inside the inner summation is from Eq. (7). It is important to note that the ground-truth target symbols $y_t^{(n)}$ are used during training. The entire model is differentiable, and the gradient of the log-likelihood function with respect to all the parameters $\boldsymbol{\theta}$ can be computed efficiently by backpropagation. This makes it straightforward to use stochastic gradient descent or its variants to train the whole model jointly to maximize the translation performance.

3 Multi-Way, Multilingual Translation

In this section, we discuss issues and our solutions in extending the conventional *single-pair* attention-based neural machine translation into *multi-way, multilingual* model.

Problem Definition We assume $N > 1$ source languages $\{X^1, X^2, \ldots, X^N\}$ and $M > 1$ target languages $\{Y^1, Y^2, \ldots, Y^M\}$, and the availability of $L \leq M \times N$ *bilingual* parallel corpora $\{D_1, \ldots, D_L\}$, each of which is a set of sentence pairs of one source and one target language. We use $s(D_l)$ and $t(D_l)$ to indicate the source and target languages of the l-th parallel corpus.

For each parallel corpus l, we can directly use the log-likelihood function from Eq. (8) to define a pair-specific log-likelihood $\mathcal{L}^{s(D_l),t(D_l)}$. Then, the goal of multi-way, multilingual neural machine translation is to build a model that maximizes the joint log-likelihood function $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{L}\sum_{l=1}^{L} \mathcal{L}^{s(D_l),t(D_l)}(\boldsymbol{\theta})$. Once the training is over, the model can do translation from any of the source languages to any of the target languages included in the parallel training corpora.

3.1 Existing Approaches

Neural Machine Translation without Attention In (Luong et al., 2015a), the authors extended the basic encoder-decoder network for multitask neural machine translation. As they extended the basic encoder-decoder network, their model effectively becomes a set of encoders and decoders, where each of the encoder projects a source sentence into a common vector space. The point in the common space is then decoded into different languages.

The major difference between (Luong et al., 2015a) and our work is that we extend the attention-based encoder-decoder instead of the basic model. This is an important contribution, as the attention-based neural machine translation has become *de facto* standard in neural translation literatures recently (Jean et al., 2014; Jean et al., 2015; Luong et al., 2015b; Sennrich et al., 2015b; Sennrich et al., 2015a), by opposition to the basic encoder-decoder.

There are two minor differences as well. First, they do not consider multilinguality in depth. The authors of (Luong et al., 2015a) tried only a single language pair, English and German, in their model. Second, they only report translation perplexity, which is not a widely used metric for measuring translation quality. To more easily compare with other machine translation approaches it would be important to evaluate metrics such as BLEU, which counts the number of matched n-grams between the generated and reference translations.

One-to-Many Neural Machine Translation The authors of (Dong et al., 2015) earlier proposed a multilingual translation model based on the *attention-based neural machine translation*. Unlike this paper, they only tried it on one-to-many translation, similarly to earlier work by (Collobert et al., 2011) where one-to-many natural language processing was done. In this setting, it is trivial to extend the single-pair attention-based model into multilingual translation by simply having a single encoder for a source language and pairs of a decoder and attention mechanism (Eq. (2)) for each target language. We will shortly discuss more on why, with the attention mechanism, one-to-many translation is trivial, while multi-way translation is not.

3.2 Challenges

A quick look at neural machine translation seems to suggest a straightforward path toward incorporating multiple languages in both source and target sides. As described earlier already in the introduction, the basic idea is simple. We assign a separate encoder to each source language and a separate decoder to each target language. The encoder will project a source sentence in its own language into a common, language-agnostic space, from which the decoder will generate a translation in its own language.

Unlike training multiple single-pair neural translation models, in this case, the encoders and decoders are shared across multiple pairs. This is computationally beneficial, as the number of parameters grows only linearly with respect to the number of languages ($O(L)$), in contrary to training separate single-pair models, in which case the number of parameters grows quadratically ($O(L^2)$.)

The attention mechanism, which was initially called a soft-alignment model in (Bahdanau et al., 2014), aligns a (potentially non-contiguous) source phrase to a target word. This alignment process is largely specific to a language pair, and it is not clear whether an alignment mechanism for one language pair can also work for another pair.

The most naive solution to this issue is to have $O(L^2)$ attention mechanisms that are *not* shared across multiple language pairs. Each attention mechanism takes care of a single pair of source and

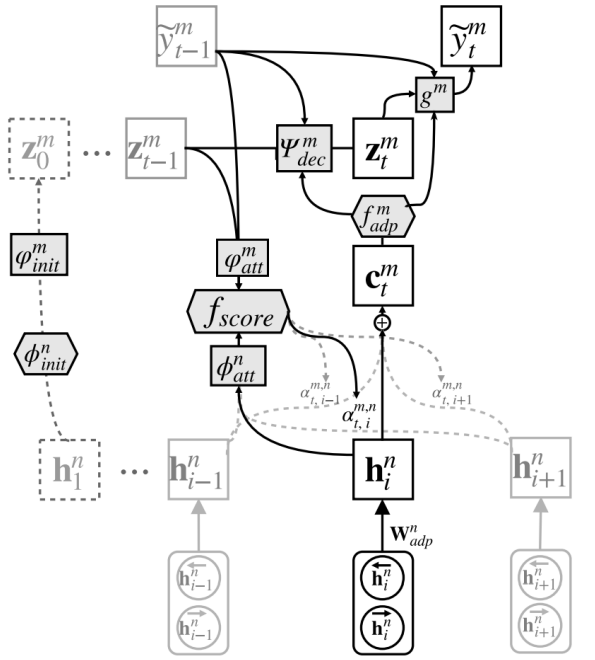

Figure 1: One step of the proposed multi-way. multilingual Neural Machine Translation model, for the n-th encoder and the m-th decoder at time step t. Shaded boxes are parametric functions and square boxes represent intermediate variables of the model. Initializer network is also illustrated as the left-most network with dashed boxes. Notice, all the shared components are drawn with diamond boxes. See Sec. 4 for details.

target languages. This is the approach employed in (Dong et al., 2015), where each decoder had its own attention mechanism.

There are two issues with this naive approach. First, unlike what has been hoped initially with multilingual neural machine translation, the number of parameters again grows quadratically w.r.t. the number of languages. Second and more importantly, having separate attention mechanisms makes it less likely for the model to fully benefit from having multiple tasks (Caruana, 1997), especially for transfer learning towards resource-poor languages.

In short, the major challenge in building a multi-way, multilingual neural machine translation is in avoiding independent (i.e., quadratically many) attention mechanisms. There are two questions behind this challenge. The first one is whether it is even possible to share a single attention mechanism across multiple language pairs. The second question immediately follows: how can we build a neural translation model to share a single attention mechanism for all the language pairs in consideration?

4 Multi-Way, Multilingual Model

We describe in this section, the proposed *multi-way, multilingual attention-based neural machine translation*. The proposed model consists of N encoders $\{\Psi^n_{\text{enc}}\}_{n=1}^N$ (see Eq. (1)), M decoders $\{(\Psi^m_{\text{dec}}, g^m, f^m_{\text{init}})\}_{m=1}^M$ (see Eqs. (5)–(7)) and a shared attention mechanism f_{score} (see Eq. (2) in the single language pair case).

Encoders Similarly to (Luong et al., 2015b), we have one encoder per source language, meaning that a single encoder is shared for translating the language to multiple target languages. In order to handle different source languages better, we may use for each source language a different type of encoder, for instance, of different size (in terms of the number of recurrent units) or of different architecture (convolutional instead of recurrent.)[1] This allows us to efficiently incorporate varying types of languages in the proposed multilingual translation model.

This however implies that the dimensionality of the context vectors in Eq. (1) may differ across source languages. Therefore, we add to the original bidirectional encoder from Sec. 2, a linear transformation layer consisting of a weight matrix $\mathbf{W}^n_{\text{adp}}$ and a bias vector $\mathbf{b}^n_{\text{adp}}$, which is used to project each context vector into a common dimensional space:

$$\mathbf{h}^n_t = \mathbf{W}^n_{\text{adp}} \left[\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t \right] + \mathbf{b}^n_{\text{adp}}, \tag{9}$$

where $\mathbf{W}^n_{\text{adp}} \in \mathbb{R}^{d \times (\dim \overrightarrow{\mathbf{h}}_t + \dim \overleftarrow{\mathbf{h}}_t)}$ and $\mathbf{b}^n_{\text{adp}} \in \mathbb{R}^d$.

In addition, each encoder exposes two transformation functions ϕ^n_{att} and ϕ^n_{init}. The first transformer ϕ^n_{att} transforms a context vector to be compatible with a shared attention mechanism:

$$\tilde{\mathbf{h}}^n_t = \phi^n_{\text{att}}(\mathbf{h}^n_t). \tag{10}$$

This transformer can be implemented as any type of parametric function, and in this paper, we simply apply an element-wise $\tanh$ to $\mathbf{h}^n_t$.

[1] For the pairs without enough parallel data, one may also consider using smaller encoders to prevent over-fitting.

The second transformer ϕ^n_{init} transforms the first context vector $\mathbf{h}^n_1$ to be compatible with the initializer of the decoder's hidden state (see Eq. (6)):

$$\hat{\mathbf{h}}^n_1 = \phi^n_{\text{init}}(\mathbf{h}^n_1). \quad (11)$$

Similarly to ϕ^n_{att}, it can be implemented as any type of parametric function. In this paper, we use a feedforward network with a single hidden layer and share one network ϕ_{init} for all encoder-decoder pairs.

Decoders We first start with an initialization of the decoder's hidden state. Each decoder has its own parametric function φ^m_{init} that maps the last context vector $\hat{\mathbf{h}}^n_{T_x}$ of the source encoder from Eq. (11) into the initial hidden state:

$$\mathbf{z}^m_0 = \varphi^m_{\text{init}}(\hat{\mathbf{h}}^n_{T_x}) = \varphi^m_{\text{init}}(\phi^n_{\text{init}}(\mathbf{h}^n_1))$$

φ^m_{init} can be any parametric function, and in this paper, we used a feedforward network with a single $\tanh$ hidden layer.

Each decoder exposes a parametric function φ^m_{att} that transforms its hidden state and the previously decoded symbol to be compatible with a shared attention mechanism. This transformer is a parametric function that takes as input the previous hidden state $\mathbf{z}^m_{t-1}$ and the previous symbol $\tilde{y}^m_{t-1}$ and returns a vector for the attention mechanism:

$$\tilde{\mathbf{z}}^m_{t-1} = \varphi^m_{\text{att}}\left(\mathbf{z}^m_{t-1}, \mathbf{E}^m_y\left[\tilde{y}^m_{t-1}\right]\right) \quad (12)$$

which replaces $\mathbf{z}_{t-1}$ in Eq. 2. In this paper, we use a feedforward network with a single $\tanh$ hidden layer for each φ^m_{att}.

Given the previous hidden state $\mathbf{z}^m_{t-1}$, previously decoded symbol $\tilde{y}^m_{t-1}$ and the time-dependent context vector $\mathbf{c}^m_t$, which we will discuss shortly, the decoder updates its hidden state:

$$\mathbf{z}_t = \Psi_{\text{dec}}\left(\mathbf{z}^m_{t-1}, \mathbf{E}^m_y\left[\tilde{y}^m_{t-1}\right], f^m_{\text{adp}}(\mathbf{c}^m_t)\right),$$

where f^m_{adp} affine-transforms the time-dependent context vector to be of the same dimensionality as the decoder. We share a single affine-transformation layer f^m_{adp}, for all the decoders in this paper.

Once the hidden state is updated, the probability distribution over the next symbol is computed exactly as for the pair-specific model (see Eq. (7).)

	# Symbols		# Sentence
	# En	Other	Pairs
En-Fr	1.022b	2.213b	38.85m
En-Cs	186.57m	185.58m	12.12m
En-Ru	50.62m	55.76m	2.32m
En-De	111.77m	117.41m	4.15m
En-Fi	52.76m	43.67m	2.03m

Table 1: Statistics of the parallel corpora from WMT'15. Symbols are BPE-based sub-words.

Attention Mechanism Unlike the encoders and decoders of which there is an instance for each language, there is only a single attention mechanism, shared across all the language pairs. This shared mechanism uses the *attention-specific* vectors $\tilde{\mathbf{h}}^n_t$ and $\tilde{\mathbf{z}}^m_{t-1}$ from the encoder and decoder, respectively.

The relevance score of each context vector $\mathbf{h}^n_t$ is computed based on the decoder's previous hidden state $\mathbf{z}^m_{t-1}$ and previous symbol $\tilde{y}^m_{t-1}$:

$$e^{m,n}_{t,i} = f_{\text{score}}\left(\tilde{\mathbf{h}}^n_t, \tilde{\mathbf{z}}^m_{t-1}, \tilde{y}^m_{t-1}\right)$$

These scores are normalized according to Eq. (3) to become the attention weights $\alpha^{m,n}_{t,i}$.

With these attention weights, the time-dependent context vector is computed as the weighted sum of the *original* context vectors: $\mathbf{c}^{m,n}_t = \sum_{i=1}^{T_x} \alpha^{m,n}_{t,i}\mathbf{h}^n_i$.

See Fig. 1 for the illustration.

5 Experiment Settings

5.1 Datasets

We evaluate the proposed multi-way, multilingual translation model on all the pairs available from WMT'15–English (En) ↔ French (Fr), Czech (Cs), German (De), Russian (Ru) and Finnish (Fi)–, totalling ten directed pairs. For each pair, we concatenate all the available parallel corpora from WMT'15 and use it as a training set. We use newstest-2013 as a development set and newstest-2015 as a test set, in all the pairs other than Fi-En. In the case of Fi-En, we use newsdev-2015 and newstest-2015 as a development set and test set, respectively.

Data Preprocessing Each training corpus is tokenized using the tokenizer script from the Moses decoder.[2] The tokenized training corpus is cleaned fol-

[2] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

lowing the procedure in (Jean et al., 2015). Instead of using space-separated tokens, or words, we use sub-word units extracted by byte pair encoding, as recently proposed in (Sennrich et al., 2015b). For each and every language, we include 30k sub-word symbols in a vocabulary. See Table 1 for the statistics of the final, preprocessed training corpora.

Evaluation Metric We mainly use BLEU as an evaluation metric using the multi-bleu script from Moses.[3] BLEU is computed on the tokenized text after merging the BPE-based sub-word symbols. We further look at the average log-probability assigned to reference translations by the trained model as an additional evaluation metric, as a way to measure the model's density estimation performance free from any error caused by approximate decoding.

5.2 Two Scenarios

Low-Resource Translation First, we investigate the effect of the proposed multi-way, multilingual model on low-resource language-pair translation. Among the six languages from WMT'15, we choose En, De and Fi as source languages, and En and De as target languages. We control the amount of the parallel corpus of each pair out of three to be 5%, 10%, 20% and 40% of the original corpus. In other words, we train four models with different sizes of parallel corpus for each language pair (En-De, De-En, Fi-En.)

As a baseline, we train a single-pair model for each multi-way, multilingual model. We further finetune the single-pair model to incorporate the target-side monolingual corpus consisting of all the target side text from the other language pairs (e.g., when a single-pair model was trained on Fi-En, the target-side monolingual corpus consists of the target sides from De-En.) This is done by the recently proposed deep fusion (Gulcehre et al., 2015). The latter is included to tell whether any improvement from the multilingual model is simply due to the increased amount of target-side monolingual corpus.

Large-scale Translation We train one multi-way, multilingual model that has six encoders and six decoders, corresponding to the six languages from WMT'15; En, Fr, De, Cs, Ru, Fi → En, Fr, De, Cs, Ru, Fi. We use the full corpora for all of them.

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

	Size	Single	Single+DF	Multi
Fi→En	100k	5.06/3.96	4.98/3.99	6.2/**5.17**
	200k	7.1/6.16	7.21/6.17	8.84/**7.53**
	400k	9.11/7.85	9.31/8.18	11.09/**9.98**
	800k	11.08/9.96	11.59/10.15	12.73/**11.28**
De→En	210k	14.27/13.2	14.65/13.88	16.96/**16.26**
	420k	18.32/17.32	18.51/17.62	19.81/**19.63**
	840k	21/19.93	21.69/20.75	22.17/**21.93**
	1.68m	23.38/23.01	23.33/22.86	23.86/**23.52**
En→De	210k	11.44/11.57	11.71/11.16	12.63/**12.68**
	420k	14.28/14.25	14.88/15.05	15.01/**15.67**
	840k	17.09/17.44	17.21/17.88	17.33/**18.14**
	1.68m	19.09/19.6	19.36/20.13	19.23/**20.59**

Table 2: BLEU scores where the target pair's parallel corpus is constrained to be 5%, 10%, 20% and 40% of the original size. We report the BLEU scores on the development and test sets (separated by /) by the single-pair model (Single), the single-pair model with monolingual corpus (Single+DF) and the proposed multi-way, multilingual model (Multi).

5.3 Model Architecture

Each symbol, either source or target, is projected on a 620-dimensional space. The encoder is a bidirectional recurrent neural network with 1,000 gated recurrent units (GRU) in each direction, and the decoder is a recurrent neural network with also 1,000 GRU's. The decoder's output function g_k from Eq. (7) is a feedforward network with 1,000 $\tanh$ hidden units. The dimensionalities of the context vector $\mathbf{h}_t^n$ in Eq. (9), the attention-specific context vector $\tilde{\mathbf{h}}_t^n$ in Eq. (10) and the attention-specific decoder hidden state $\tilde{\mathbf{h}}_{t-1}^m$ in Eq. (12) are all set to 1,200.

We use the same type of encoder for every source language, and the same type of decoder for every target language. The only difference between the single-pair models and the proposed multilingual ones is the numbers of encoders N and decoders M. We leave those multilingual translation specific components, such as the ones in Eqs. (9)–(12), in the single-pair models in order to keep the number of shared parameters constant.

5.4 Training

Basic Settings We train each model using stochastic gradient descent (SGD) with Adam (Kingma and

		Dir	Fr (39m) → En	Fr (39m) En →	Cs (12m) → En	Cs (12m) En →	De (4.2m) → En	De (4.2m) En →	Ru (2.3m) → En	Ru (2.3m) En →	Fi (2m) → En	Fi (2m) En →
(a) BLEU	Dev	Single	27.22	26.91	21.24	15.9	24.13	20.49	21.04	18.06	13.15	9.59
		Multi	26.09	25.04	21.23	14.42	23.66	19.17	21.48	17.89	12.97	8.92
	Test	Single	27.94	**29.7**	20.32	**13.84**	24	**21.75**	22.44	**19.54**	12.24	**9.23**
		Multi	**28.06**	27.88	**20.57**	13.29	**24.20**	20.59	**23.44**	19.39	**12.61**	8.98
(b) LL	Dev	Single	-50.53	-53.38	-60.69	-69.56	-54.76	-61.21	-60.19	-65.81	-88.44	-91.75
		Multi	-50.6	-56.55	-54.46	-70.76	-54.14	-62.34	-54.09	-63.75	-74.84	-88.02
	Test	Single	-43.34	**-45.07**	-60.03	**-64.34**	-57.81	**-59.55**	-60.65	-60.29	-88.66	-94.23
		Multi	**-42.22**	-46.29	**-54.66**	-64.80	**-53.85**	-60.23	**-54.49**	**-58.63**	**-71.26**	**-88.09**

Table 3: (a) BLEU scores and (b) average log-probabilities for all the five languages from WMT'15.

Ba, 2015) as an adaptive learning rate algorithm. We use the initial learning rate of $2 \cdot 10^{-4}$ and leave all the other hyperparameters as suggested in (Kingma and Ba, 2015). Each SGD update is computed using a minibatch of 80 examples, unless the model is parallelized over two GPUs, in which case we use a minibatch of 60 examples. We only use sentences of length up to 50 symbols during training. We clip the norm of the gradient to be no more than 1 (Pascanu et al., 2012). All training runs are early-stopped based on BLEU on the development set. As we observed in preliminary experiments better scores on the development set when finetuning the shared parameters and output layers of the decoders in the case of multilingual models, we do this for all the multilingual models. During finetuning, we clip the norm of the gradient to be no more than 5. [4]

Schedule As we have access only to bilingual corpora, each sentence pair updates only a subset of the parameters. Excessive updates based on a single language pair may bias the model away from the other pairs. To avoid it, we cycle through all the language pairs, one pair at a time, in Fi⇆En, De⇆En, Fr⇆En, Cs⇆En, Ru⇆En order. [5] Initial experiments on random scheduling across pairs and increasing the number of consecutive updates for a pair did not give better results and left as a future work.

Model Parallelism The size of the multilingual model grows linearly w.r.t. the number of languages. We observed that a single model that handles six source and six target languages does not fit in a single GPU[6] during training. We address this by distributing computational paths according to different translation pairs over multiple GPUs, following (Ding et al., 2014). The shared parameters, mainly related to the attention mechanism, is duplicated on both GPUs.

In more detail, we distribute language pairs across multiple GPUs such that those pairs in each GPU shares either an encoder or decoder. This allows us to avoid synchronizing a large subset of the parameters across multiple GPUs. Only the shared attention mechanism, which has substantially less parameters, is duplicated on all the GPUs. Before each update, we build a minibatch to contain an approximately equal number of examples per GPU in order to minimize any discrepancy in computation among multiple GPUs. Each GPU then computes the gradient w.r.t. the parameters on its own board and updates the local parameters. The gradients w.r.t. the attention mechanism are synchronized using direct memory access (DMA). In this way, we achieve near-linear speed-up.

6 Results and Analysis

Low-Resource Translation It is clear from Table 2 that the proposed model (Multi) outperforms the single-pair one (Single) in all the cases. This is true even when the single-pair model is strengthened with a target-side monolingual corpus (Single+DF). This suggests that the benefit of generalization from having multiple languages goes beyond that of simply having more target-side monolingual corpus. The performance gap grows as the size of target parallel corpus decreases.

[4] All the training details as well as the code is available at `http://github.com/nyu-dl/dl4mt-multi`.

[5] ⇆ indicates simultaneous updates on two GPUs.

[6] NVidia Titan X with 12GB on-board memory

Further, directly adding monolingual data from all languages during training, e.g. like an auto-encoder, En→ En, De→ De etc. is straightforward. In fact, experiments based on the autoencoder reconstruction criterion resulted in rapid memorization, copying source tokens without capturing semantics, resulting in worse performance. Exploring ways to leverage unlabeled data and regularizing the monolingual paths in the multi-way, multilingual architecture, is therefore left as a future work.

Large-Scale Translation In Table 3, we observe that the proposed multilingual model outperforms or is comparable to the single-pair models for the majority of the all ten pairs/directions considered. This happens in terms of both BLEU and average log-probability. This is encouraging, considering that there are twice more parameters in the whole set of single-pair models than in the multilingual model.

Note that, the numbers are below state-of-the-art neural MT systems, which use large vocabularies, unknown replacements techniques and ensembling. We mainly focused on comparing the proposed model against single-pair models without these techniques in order to carefully control and analyze the effect of having multiple languages. It is indeed required in the future to analyze the consequence of having both multiple languages and other such techniques in a single model.

It is worthwhile to notice that the benefit is more apparent when the model translates from a foreign language to English. This may be due to the fact that all of the parallel corpora include English as either a source or target language, leading to a better parameter estimation of the English decoder. In the future, a strategy of using a pseudo-parallel corpus to increase the amount of training examples for the decoders of other languages (Sennrich et al., 2015a) should be investigated to confirm this conjecture.

7 Conclusion

In this paper, we proposed multi-way, multilingual attention-based neural machine translation. The proposed approach allows us to build a single neural network that can handle multiple source and target languages simultaneously. The proposed model is a step forward from the recent works on multilingual neural translation, in the sense that we support attention mechanism, compared to (Luong et al., 2015a) and multi-way translation, compared to (Dong et al., 2015). Furthermore, we evaluate the proposed model on large-scale experiments, using the full set of parallel corpora from WMT'15.

We empirically evaluate the proposed model in large-scale experiments using all five languages from WMT'15 with the full set of parallel corpora and also in the settings with artificially controlled amount of the target parallel corpus. In both of the settings, we observed the benefits of the proposed multilingual neural translation model over having a set of single-pair models. The improvement was especially clear in the cases of translating low-resource language pairs.

We observed the larger improvements when translating to English. We conjecture that this is due to a higher availability of English in most parallel corpora, leading to a better parameter estimation of the English decoder. More research on this phenomenon in the future will result in further improvements from using the proposed model. Also, all the other techniques proposed recently, such as ensembling and large vocabulary tricks, need to be tried together with the proposed multilingual model to improve the translation quality even further. Finally, an interesting future work is to use the proposed model to translate between a language pair not included in a set of training corpus.

Acknowledgments

We acknowledge the support of the following agencies for research funding and computing support: NSERC, Calcul Québec, Compute Canada, the Canada Research Chairs, CIFAR and Samsung. OF thanks the support by TUBITAK (2214/A). KC thanks the support by Facebook and Google (Google Faculty Award 2016).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*.

Léon Bottou, Yoshua Bengio, and Yann Le Cun. 1997. Global training of document processing systems using graph transformer networks. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997*

IEEE Computer Society Conference on, pages 489–494. IEEE.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–Decoder approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, October.

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

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in Neural Information Processing Systems*, pages 577–585.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Weiguang Ding, Ruoyan Wang, Fei Mao, and Graham W. Taylor. 2014. Theano-based large-scale visual recognition with multiple gpus. *arXiv:1412.2302*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. ACL.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM.

Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. 2009. A novel connectionist system for unconstrained handwriting recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):855–868.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

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

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. In *ACL 2015*.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for wmt'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, Lisbon, Portugal, September. Association for Computational Linguistics.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *The International Conference on Learning Representations (ICLR)*.

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

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

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*.

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

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

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