Contextual Parameter Generation for Universal Neural Machine Translation

Emmanouil Antonios Platanios[†], **Mrinmaya Sachan**[†], **Graham Neubig**[‡], **Tom M. Mitchell**[†]
[†]Machine Learning Department, [‡]Language Technologies Institute
Carnegie Mellon University
{e.a.platanios,mrinmays,gneubig,tom.mitchell}@cs.cmu.edu

Abstract

We propose a simple modification to existing neural machine translation (NMT) models that enables using a single *universal* model to translate between multiple languages while allowing for language specific parameterization, and that can also be used for *domain adaptation*. Our approach requires no changes to the model architecture of a standard NMT system, but instead introduces a new component, the *contextual parameter generator* (CPG), that generates the parameters of the system (e.g., weights in a neural network). This parameter generator accepts source and target language embeddings as input, and generates the parameters for the encoder and the decoder, respectively. The rest of the model remains unchanged and is shared across all languages. We show how this simple modification enables the system to use monolingual data for training and also perform *zero-shot* translation. We further show it is able to surpass state-of-the-art performance for both the IWSLT-15 and IWSLT-17 datasets and that the learned language embeddings are able to uncover interesting relationships between languages.

1 Introduction

Neural Machine Translation (NMT) directly models the mapping of a source language to a target language without any need for training or tuning any component of the system separately. This has led to a rapid progress in NMT and its successful adoption in many large-scale settings (Wu et al., 2016; Crego et al., 2016). The encoder-decoder abstraction makes it conceptually feasible to build a system that maps any source sentence in any language to a vector representation, and then decodes this representation into any target language. Thus, various approaches have been proposed to extend this abstraction for multilingual MT (Luong et al., 2016; Dong et al., 2015; Johnson et al., 2017; Ha et al., 2016; Firat et al., 2016a).

Prior work in multilingual NMT can be broadly categorized into two paradigms. The first, *universal NMT* (Johnson et al., 2017; Ha et al., 2016), uses a single model for all languages. Universal NMT lacks any language-specific parameterization, which is an oversimplification and detrimental when we have very different languages and limited training data. As verified by our experiments, the method of Johnson et al. (2017) suffers from high sample complexity and thus underperforms in limited data settings. The universal model proposed by Ha et al. (2016) requires a new coding scheme for the input sentences, which results in large vocabulary sizes that are difficult to scale. The second paradigm, *per-language encoder-decoder* (Luong et al., 2016; Firat et al., 2016a), uses separate encoders and decoders for each language. This does not allow for sharing of information across languages, which can result in overparameterization and can be detrimental when the languages are similar.

In this paper, we strike a balance between these two approaches, proposing a model that has the ability to learn parameters separately for each language, but also share information between similar languages. We propose using a new *contextual parameter generator* (CPG) which (a) generalizes all of these methods, and (b) mitigates the aforementioned issues of *universal* and *per-language encoder-decoder* systems. It learns language embeddings as a context for translation and uses them to generate the parameters of a shared translation model for all language pairs. Thus, it provides these models the ability to learn parameters separately for each language, but also share information between similar languages. The parameter generator is general and allows any existing NMT model to be enhanced in this way.[1] In addition, it has the following desirable features:

1. **Simple:** Similar to Johnson et al. (2017) and Ha et al. (2016), and in contrast with Luong et al. (2016) and Firat et al. (2016a), it can

[1] In fact, it could likely be applied in other scenarios, such as domain adaptation, as well.

be applied to most existing NMT systems with some minor modification, and it is able to accommodate attention layers seamlessly.
2. **Multilingual:** Enables multilingual translation using the same single model as before.
3. **Semi-supervised:** Can use monolingual data.
4. **Scalable:** Reduces the number of parameters by employing extensive, yet controllable, sharing across languages, thus mitigating the need for large amounts of data, as in Johnson et al. (2017). It also allows for the decoupling of languages, avoiding the need for a large shared vocabulary, as in Ha et al. (2016).
5. **Adaptable:** Can adapt to support new languages, without requiring complete retraining.
6. **State-of-the-art:** Achieves better performance than pairwise NMT models and Johnson et al. (2017). In fact, our approach can surpass state-of-the-art performance.

We first introduce a modular framework that can be used to define and describe most existing NMT systems. Then, in Section 3, we introduce our main contribution, the *contextual parameter generator* (CPG), in terms of that framework. We also argue that the proposed approach takes us a step closer to a common universal interlingua.

2 Background

We first define the multi-lingual NMT setting and then introduce a modular framework that can be used to define and describe most existing NMT systems. This will help us distill previous contributions and introduce ours.

Setting. We assume that we have a set of source languages S and a set of target languages T. The total number of languages is $L = |S \cup T|$. We also assume we have a set of $C \leq |S| \times |T|$ pairwise parallel corpora, $\{P_1, \ldots, P_C\}$, each of which contains a set of sentence pairs for a single source-target language combination. The goal of multilingual NMT is to build a model that, when trained using the provided parallel corpora, can learn to translate well between any pair of languages in $S \times T$. The majority of related work only considers pairwise NMT, where $|S| = |T| = 1$.

2.1 NMT Modules

Most NMT systems can be decomposed to the following modules (also visualized in Figure 1).

Preprocessing Pipeline. The data preprocessing pipeline handles tokenization, cleaning, normalizing the text data and building a *vocabulary*, i.e. a two-way mapping from preprocessed sentences to sequences of word indices that will be used for the translation. A commonly used proposal for defining the vocabulary is the byte-pair encoding (BPE) algorithm which generates subword unit vocabularies (Sennrich et al., 2016b). This eliminates the notion of out-of-vocabulary words, often resulting in increased translation quality.

Encoder/Decoder. The *encoder* takes in indexed source language sentences, and produces an intermediate representation that can later be used by a *decoder* to generate sentences in a target language. Generally, we can think of the encoder as a function, $\mathbf{f}^{(enc)}$, parameterized by $\theta^{(enc)}$. Similarly, we can think of the decoder as another function, $\mathbf{f}^{(dec)}$, parameterized by $\theta^{(dec)}$. The goal of learning to translate can then be defined as finding the values for $\theta^{(enc)}$ and $\theta^{(dec)}$ that result in the best translations. A large amount of previous work proposes novel designs for the encoder/decoder module. For example, using attention over the input sequence while decoding (Bahdanau et al., 2015; Luong et al., 2015) provides significant gains in translation performance.[2]

Parameter Generator. All modules defined so far have previously been used when describing NMT systems and are thus easy to conceptualize. However, in previous work, most models are trained for a given language pair, and it is not trivial to extend them to work for multiple pairs of languages. We introduce here the concept of the *parameter generator*, which makes it easy to define and describe multilingual NMT systems. This module is responsible for generating $\theta^{(enc)}$ and $\theta^{(dec)}$ for any given source and target language. Different parameter generators result in different numbers of learnable parameters and can thus be used to share information across different languages. Next, we describe related work, in terms of the parameter generator for NMT:

- **Pairwise:** In the simple and commonly used pairwise NMT setting (Wu et al., 2016; Crego et al., 2016), the parameter generator would generate separate parameters, $\theta^{(enc)}$ and $\theta^{(dec)}$, for each pair of source-target languages. This re-

[2]Note that depending on the vocabulary that is used and on whether it is one shared vocabulary across all languages, or one vocabulary per language, the output projection layer of the decoder (which produces probabilities over words) may be language dependent, or common across all languages. In our experiments, we used separate vocabularies and thus this layer was language-dependent.

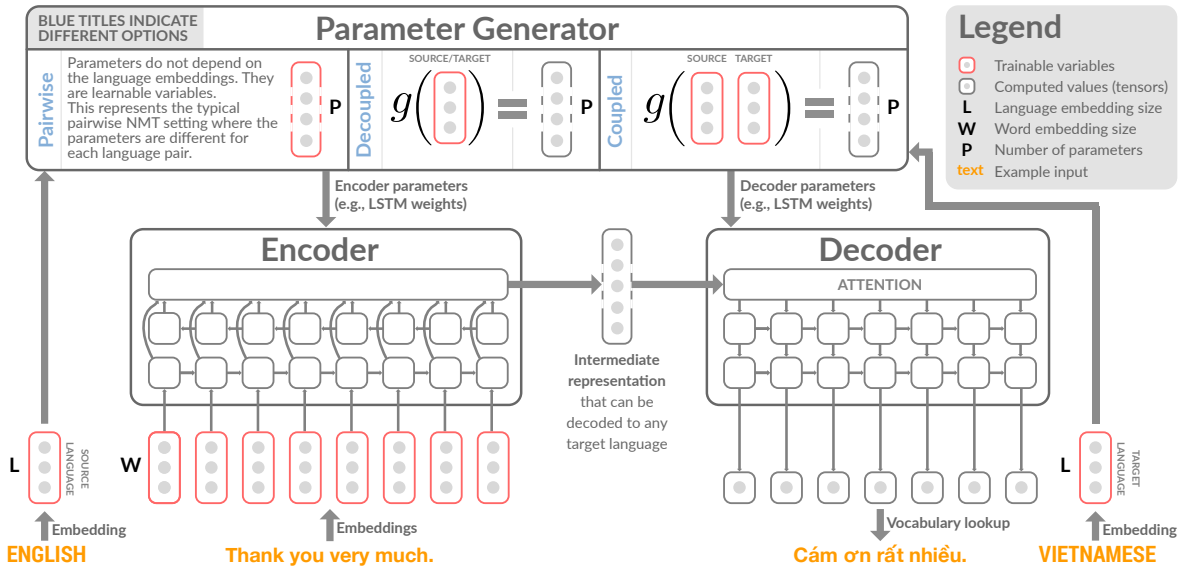

Figure 1: Overview of an NMT system, under our modular framework. Our main contribution lies in the parameter generator module (i.e., coupled or decoupled — each of the boxes with blue titles is a separate option). Note that g denotes a parameter generator network. In our experiments, we consider linear forms for this network. However, our contribution does not depend on the choices made regarding the rest of the modules; we could still use our parameter generator with different architectures for the encoder and the decoder, as well as using different kinds of vocabularies.

sults in no parameter sharing across languages, and thus $\mathcal{O}(ST)$ parameters.

- **Per-Language:** In the case of Dong et al. (2015), Luong et al. (2016) and Firat et al. (2016a), the parameter generator would generate separate encoder parameters, $\theta^{(enc)}$, for each source language, and separate decoder parameters, $\theta^{(dec)}$, for each target language. This leads to a reduction in the number of learnable parameters for multilingual NMT, from $\mathcal{O}(ST)$ to $\mathcal{O}(S+T)$. On one hand, Dong et al. (2015) train multiple models as a one-to-many multilingual NMT system that translates from one source language to multiple target languages. On the other hand, Luong et al. (2016) and Firat et al. (2016a) perform many-to-many translation. Luong et al. (2016), however, only report results for a single language pair and do not attempt multilingual translation. Firat et al. (2016a) propose an attention mechanism that is shared across all language pairs. We generalize the idea of multi-way multilingual NMT with the parameter generator network, described later.
- **Universal:** In the case of Ha et al. (2016) and Johnson et al. (2017), the authors propose using a single common set of encoder-decoder parameters for all language pairs. While Ha et al. (2016) embed words in a common semantic space across languages, Johnson et al. (2017) learn language embeddings that are in the same space as the word embeddings. Here, the parameter generator would provide the same parameters $\theta^{(enc)}$ and $\theta^{(dec)}$ for all language pairs. It would also create and keep track of learnable variables representing language embeddings that are prepended to the encoder input sequence. As we observed in our experiments, this system fails to perform well when the training data is limited. Finally, we believe that embedding languages in the same space as words is not intuitive; in our approach, languages are embedded in a separate space.

In contrast to all these related systems, we provide a simple, efficient, yet effective alternative – a parameter generator for multilingual NMT, that enables semi-supervised and zero-shot learning. We also learn language embeddings, similar to Johnson et al. (2017), but in our case they are separate from the word embeddings and are treated as a *context* for the translation, in a sense that will become clear in the next section. This notion of *context* is used to define parameter sharing across various encoders and decoders, and, as we discuss in our conclusion, is even applicable beyond NMT.

3 Proposed Method

We propose a new way to share information across different languages and to control the amount of sharing, through the parameter generator module. More specifically, we propose *contextual parameter generators*.

Contextual Parameter Generator. Let us denote the source language for a given sentence pair

by ℓ_s and the target language by ℓ_t. Then, when using the contextual parameter generator, the parameters of the encoder are defined as $\theta^{(enc)} \triangleq g^{(enc)}(\mathbf{l_s})$, for some function $g^{(enc)}$, where $\mathbf{l_s}$ denotes a language embedding for the source language ℓ_s. Similarly, the parameters of the decoder are defined as $\theta^{(dec)} \triangleq g^{(dec)}(\mathbf{l_t})$ for some function $g^{(dec)}$, where $\mathbf{l_t}$ denotes a language embedding for the target language ℓ_t. Our generic formulation does not impose any constraints on the functional form of $g^{(enc)}$ and $g^{(dec)}$. In this case, we can think of the source language, ℓ_s, as a context for the encoder. The parameters of the encoder depend on its context, but its architecture is common across all contexts. We can make a similar argument for the decoder, and that is where the name of this parameter generator comes from. We can even go a step further and have a parameter generator that defines $\theta^{(enc)} \triangleq g^{(enc)}(\mathbf{l_s}, \mathbf{l_t})$ and $\theta^{(dec)} \triangleq g^{(dec)}(\mathbf{l_s}, \mathbf{l_t})$, thus coupling the encoding and decoding stages for a given language pair. In our experiments we stick to the previous, *decoupled*, form, because unlike Johnson et al. (2017), it has the potential to lead to an *interlingua*.

Concretely, because the encoding and decoding stages are decoupled, the encoder is not aware of the target language while generating it. Thus, we can take an encoded intermediate representation of a sentence and translate it to any target language. This is because, in this case, the intermediate representation is independent of any target language. This makes for a stronger argument that the intermediate representation produced by our encoder could be approaching a universal interlingua, more so than methods that are aware of the target language when they perform encoding.

3.1 Parameter Generator Network

We refer to the functions $g^{(enc)}$ and $g^{(dec)}$ as *parameter generator networks*. Even though our proposed NMT framework does not rely on a specific choice for $g^{(enc)}$ and $g^{(dec)}$, here we describe the functional form we used for our experiments. Our goal is to provide a simple form that works, and for which we can reason about. For this reason, we decided to define the parameter generator networks as simple linear transforms, similar to the factored adaptation model of Michel and Neubig (2018), which was only applied to the bias terms of the output softmax:

$$g^{(enc)}(\mathbf{l_s}) \triangleq \mathbf{W^{(enc)}}\mathbf{l_s}, \quad (1)$$

$$g^{(dec)}(\mathbf{l_t}) \triangleq \mathbf{W^{(dec)}}\mathbf{l_t}, \quad (2)$$

where $\mathbf{l_s}, \mathbf{l_t} \in \mathbb{R}^{\mathbf{M}}$, $\mathbf{W^{(enc)}} \in \mathbb{R}^{\mathbf{P^{(enc)}} \times \mathbf{M}}$, $\mathbf{W^{(dec)}} \in \mathbb{R}^{\mathbf{P^{(dec)}} \times \mathbf{M}}$, M is the language embedding size, $P^{(enc)}$ is the number of parameters of the encoder, and $P^{(dec)}$ is the number of parameters of the decoder.

Another way to interpret this model is that it imposes a low-rank constraint on the parameters. As opposed to our approach, in the base case of using multiple pairwise models to perform multilingual translation, each model has $P = P^{(enc)} + P^{(dec)}$ learnable parameters for its encoder and decoder. Given that the models are pairwise, for L languages, we have a total of $L(L-1)$ learnable parameter vectors of size P. On the other hand, using our contextual parameter generator we have a total of L vectors of size M (one for each language), and a single matrix of size $P \times M$. Then, the parameters of the encoder and the decoder, for a single language pair, are defined as a linear combination of the M columns of that matrix.

Controlled Parameter Sharing. We can further control parameter sharing by observing that the encoder/decoder parameters often have some "natural grouping". For example, in the case of recurrent neural networks we may have multiple weight matrices, one for each layer, as well as attention-related parameters. Based on this observations, we now propose a way to control how much information is shared across languages. The language embeddings need to represent all of the language-specific information and thus may need to be large in size. However, when computing the parameters of each group, only a small part of that information is relevant. Let $\theta^{(enc)} = \{\theta_j^{(enc)}\}_{j=1}^{G}$ and $\theta_j^{(enc)} \in \mathbb{R}^{P_j^{(enc)}}$, where G denotes the number of groups. Then, we define:

$$\theta_j^{(enc)} \triangleq \mathbf{W_j^{(enc)}}\mathbf{P_j^{(enc)}}\mathbf{l_s}, \quad (3)$$

where $\mathbf{W_j^{(enc)}} \in \mathbb{R}^{P_j^{(enc)} \times M'}$ and $\mathbf{P_j^{(enc)}} \in \mathbb{R}^{M' \times M}$, with $M' < M$ (and similarly for the decoder parameters). We can see now that $\mathbf{P_j^{(enc)}}$ is used to extract the relevant information (size M') for parameter group j, from the larger language embedding (size M). This allows us to control the parameter sharing across languages in the following way: if we want to increase the number of per-language parameters (i.e., the language embedding size) we can increase M while keeping M' small enough so that the total number of parameters does not explode. This would not have been possible without the proposed low-rank ap-

proximation for $\mathbf{W}^{(\mathbf{enc})}$, that uses the parameter grouping information.

Alternative Options. Given that our proposed approach does not depend on the specific choice of the parameter generator network, it might be interesting to design models that use side-information about the languages that are being used (such as linguistic information about language families and hierarchies). This is outside the scope of this paper, but may be an interesting future direction.

3.2 Semi-Supervised and Zero-Shot Learning

The proposed parameter generator also enables semi-supervised learning via back-translation. Concretely, monolingual data can be used to train the shared encoder/decoder networks to translate a sentence from some language to itself (similar to the idea of auto-encoders by Vincent et al. (2008)). This is possible and can help learning because of the fact that many of the learnable parameters are shared across languages.

Furthermore, zero-shot translation, where the model translates between language pairs for which it has seen no explicit training data, is also possible. This is because the same per-language parameters are used to translate to and from a given language, irrespective of the language at the other end. Therefore, as long as we train our model using some language pairs that involve a given language, it is possible to learn to translate in any direction involving that language.

3.3 Potential for Adaptation

Let us assume that we have trained a model using data for some set of languages, $\ell_1, \ell_2, \ldots, \ell_m$. If we obtain data for some new language ℓ_n, we do not have to retrain the whole model from scratch. In fact, we can fix the parameters that are shared across all languages and only learn the embedding for the new language (along with the relevant word embeddings if not using a shared vocabulary). Assuming that we had a sufficient number of languages in the beginning, this may allow us to obtain reasonable translation performance for the new language, with a minimal amount of training.[3]

3.4 Number of Parameters

For the base case of using multiple pairwise models to perform multilingual translation, each model has $P + 2WV$ parameters, where $P = P^{(enc)} + P^{(dec)}$, W is the word embedding size, and V is the vocabulary size per language (assumed to be the same across languages, without loss of generality). Given that the models are pairwise, for L languages, we have a total of $L(L-1)(P + 2WV) = \mathcal{O}(L^2P + 2L^2WV)$ learnable parameters. For our approach, using the linear parameter generator network presented in Section 3.1, we have a total of $\mathcal{O}(PM + LWV)$ learnable parameters. Note that the number of encoder/decoder parameters has no dependence on L now, meaning that our model can easily scale to a large number of languages.

4 Experiments

In this section, we describe our experimental setup along with our results and key observations.

Setup. For all our experiments we use as the base NMT model an encoder-decoder network which uses a bidirectional LSTM for the encoder, and a two-layer LSTM with the attention model of Bahdanau et al. (2015) for the decoder. The word embedding size is set to 512. This is a common baseline model that achieves reasonable performance and we decided to use it as-is, without tuning any of its parameters, as extensive hyperparameter search is outside the scope of this paper.

During training, we use a label smoothing factor of 0.1 (Wu et al., 2016) and the AMSGrad optimizer (Reddi et al., 2018) with its default parameters in TensorFlow, and a batch size of 128 (due to GPU memory constraints). Optimization was stopped when the validation set BLEU score was maximized. The order in which language pairs are used while training was as follows: we always first sample a language pair (uniformly at random), and then sample a batch for that pair (uniformly at random).[4] During inference, we employ beam search with a beam size of 10 and the length normalization scheme of (Wu et al., 2016). We want to emphasize that we did not run experiments with other architectures or configurations, and thus this architecture was not chosen because it was favorable to our method, but rather because it was a frequently mentioned baseline in existing literature.

All experiments were run on a machine with a single Nvidia V100 GPU, and 24 GBs of system memory. Our most expensive experiment took about 10 hours to complete, which would

[3] This is due to the small number of parameters that need to be learned in this case. To put this into perspective, in most of our experiments we used language embeddings of size 8.

[4] We did not observe any "forgetting" effect, because we keep "re-visiting" all language pairs throughout training. It would be interesting to explore other sampling schemes, but it is outside the scope of this paper.

Table 1: Comparison of our proposed approach (shaded rows) with the base pairwise NMT model (PNMT) and the Google multilingual NMT model (GML) for the IWSLT-15 dataset. The *Percent Parallel* row shows what portion of the parallel corpus is used while training; the rest is being used only as monolingual data. Results are shown for the BLEU and Meteor metrics. CPG* represents the same model as CPG, but trained without using auto-encoding training examples. The best score in each case is shown in **bold**.

		BLEU				**Meteor**			
		PNMT	GML	CPG*	CPG	PNMT	GML	CPG*	CPG
100% Parallel Data	En→Cs	14.89	15.92	16.88	**17.22**	19.72	20.93	21.51	**21.72**
	Cs→En	24.43	25.25	26.44	**27.37**	27.29	27.46	28.16	**28.52**
	En→De	25.99	15.92	26.41	**26.77**	44.72	42.97	45.97	**46.30**
	De→En	30.93	29.60	31.24	**31.77**	30.73	29.90	30.95	**31.13**
	En→Fr	38.25	34.40	38.10	**38.32**	57.43	53.86	57.42	**57.68**
	Fr→En	37.40	35.14	37.11	**37.89**	34.83	33.14	34.34	**34.89**
	En→Th	23.62	22.22	26.03	**26.33**	-	-	-	-
	Th→En	15.54	14.03	16.54	**26.77**	21.58	21.02	22.78	**23.05**
	En→Vi	27.47	25.54	28.33	**29.03**	-	-	-	-
	Vi→En	24.03	23.19	25.91	**26.38**	27.59	26.96	28.23	**28.79**
	Mean	26.26	24.12	27.30	**27.80**	32.98	32.03	33.67	**34.01**
10% Parallel Data	En→Cs	5.71	8.18	8.40	**9.49**	12.18	14.97	15.25	**15.90**
	Cs→En	6.64	14.56	14.81	**15.38**	13.02	20.04	19.98	**20.87**
	En→De	11.70	14.60	15.09	**16.03**	29.98	33.74	34.88	**36.19**
	De→En	18.10	19.02	19.77	**20.25**	22.57	23.27	23.65	**24.40**
	En→Fr	24.47	25.15	24.00	**25.79**	44.10	44.84	44.95	**46.22**
	Fr→En	23.79	25.02	24.55	**27.12**	26.28	26.61	26.20	**28.18**
	En→Th	7.86	15.58	**18.41**	17.65	-	-	-	-
	Th→En	7.13	9.11	**10.19**	10.14	13.91	16.32	16.78	**16.92**
	En→Vi	18.01	17.51	**18.92**	18.90	-	-	-	-
	Vi→En	6.69	16.00	16.28	**16.86**	13.39	21.01	21.34	**22.28**
	Mean	13.01	16.47	17.04	**17.76**	21.93	25.10	25.38	**26.37**
1% Parallel Data	En→Cs	0.49	1.25	1.57	**2.38**	4.60	6.24	6.28	**8.38**
	Cs→En	1.10	1.76	1.87	**4.60**	6.29	7.13	7.08	**11.15**
	En→De	1.22	4.13	4.06	**6.46**	12.23	18.29	17.61	**23.83**
	De→En	1.46	3.42	3.86	**7.49**	7.58	8.79	8.95	**13.73**
	En→Fr	2.88	7.74	7.41	**12.45**	13.88	21.29	21.80	**30.36**
	Fr→En	4.05	5.22	5.06	**11.39**	9.58	9.86	9.83	**16.34**
	En→Th	1.22	5.72	8.01	**9.26**	-	-	-	-
	Th→En	1.42	1.66	1.65	**3.37**	6.08	7.22	5.89	**8.74**
	En→Vi	5.35	5.61	5.48	**8.00**	-	-	-	-
	Vi→En	2.01	3.57	3.64	**6.43**	7.86	8.76	8.48	**12.04**
	Mean	2.12	4.01	4.26	**7.18**	8.51	10.95	10.74	**15.58**

cost about $25 on a cloud computing service such as Google Cloud or Amazon Web Services, thus making our results reproducible, even by independent researchers.

Experimental Settings. The goal of our experiments is to show how, by using a simple modification of this model, (i) we can achieve significant improvements in performance, while at the same time (ii) being more data and computation efficient, and (iii) enabling support for zero-shot translation. To that end, we perform three types of experiments:

1. Supervised: In this experiment, we use full parallel corpora to train our models. Plain pairwise NMT models (PNMT) are compared to the same models modified to use our proposed decoupled parameter generator. We use two variants: (i) one which does not use auto-encoding of monolingual data while training (CPG*), and (ii) one which does (CPG). Please refer to Section 3.2 for more details.
2. Low-Resource: Similar to the supervised experiments except that we limit the size of the parallel corpora used in training. However, for GML and CPG the full monolingual corpus is used for auto-encoding training.
3. Zero-Shot: In this experiment, our goal is to evaluate how well a model can learn to translate between language pairs that it has not seen while training. For example, a model trained using parallel corpora between English and German, and English and French, will be evaluated in translating from German to French. PNMT can perform zero-shot translation in this setting using pivoting. This means that, in the previous example, we would first translate from German to English and then from English to French (using two pairwise models for a single translation). However, pivoting is prone to error propagation incurred when chaining multiple imperfect translations. The proposed CPG

Table 2: Comparison of our proposed approach (shaded rows) with the base pairwise NMT model (PNMT) and the Google multilingual NMT model (GML) for the IWSLT-17 dataset. Results are shown for the BLEU metric only because Meteor does not support `It`, `Nl`, and `Ro`. CPG^8 represents CPG using language embeddings of size 8. The "$_{\text{C4}}$" subscript represents the low-rank version of CPG for controlled parameter sharing (see Section 3.1), using rank 4, etc. The best score in each case is shown in **bold**.

		BLEU							
		PNMT	GML	CPG^8	CPG^8_{C4}	CPG^8_{C2}	CPG^8_{C1}	$\text{CPG}^{64}_{\text{C8}}$	$\text{CPG}^{512}_{\text{C8}}$
Supervised	De→En	21.78	21.25	**22.56**	20.78	22.09	21.23	21.50	22.38
	De→It	13.16	13.84	**14.73**	14.34	14.43	13.84	14.34	14.11
	De→Ro	10.85	11.95	12.24	12.37	**12.72**	10.37	11.32	11.94
	En→De	**19.75**	17.06	19.41	19.04	18.42	17.04	17.46	19.29
	En→It	27.70	25.74	27.57	27.11	**28.21**	26.26	27.26	27.48
	En→Nl	24.41	22.46	24.47	**25.15**	24.64	23.94	24.48	24.50
	En→Ro	19.23	18.60	20.83	**20.96**	18.69	17.23	20.20	20.86
	It→De	14.39	12.76	14.61	**15.06**	14.15	13.12	14.18	14.69
	It→En	29.84	27.96	**30.62**	30.10	29.44	29.22	29.56	30.18
	It→Nl	16.74	16.27	17.99	**18.11**	18.05	17.13	17.71	17.99
	Nl→En	26.30	24.78	26.31	26.17	25.74	26.15	**26.33**	26.20
	Nl→It	16.03	16.10	16.81	**17.50**	17.03	16.81	16.89	17.09
	Nl→Ro	12.84	12.48	14.01	**14.44**	12.56	11.79	12.38	13.66
	Ro→De	12.75	12.21	13.58	**13.66**	13.02	12.62	12.96	13.63
	Ro→En	24.33	22.88	23.83	23.88	24.20	23.58	**24.65**	23.57
	Ro→Nl	13.70	14.11	15.34	**15.51**	15.11	14.65	15.29	15.19
	Mean	18.99	18.15	19.68	**19.75**	19.28	18.44	19.16	19.74
Zero-Shot	De→Nl	12.75	12.50	12.74	**12.80**	11.65	12.41	12.67	12.75
	It→Ro	9.97	9.57	10.57	10.17	10.42	9.65	**10.69**	10.32
	Nl→De	11.32	10.47	11.52	11.20	11.28	10.89	**11.63**	11.45
	Ro→It	11.69	10.82	11.51	11.40	11.66	11.42	**11.78**	11.27
	Mean	11.43	10.84	11.59	11.39	11.25	11.09	**11.69**	11.44

models inherently support zero-shot translation and require no pivoting.

For the experiments using the CPG model without controlled parameter sharing, we use language embeddings of size 8. This is based merely on the fact that this is the largest model size we could fit on one GPU. Whenever possible, we compare against PNMT, GML by Johnson et al. (2017),[5] and other state-of-the-art results.

Datasets. We use the following datasets:

- IWSLT-15: Used for supervised and low-resource experiments only (this dataset does not support zero-shot learning). We report results for Czech (`Ch`), English (`En`), French (`Fr`), German (`De`), Thai (`Th`), and Vietnamese (`Vi`). This dataset contains ~90,000-220,000 training sentence pairs (depending on the language pair), ~500-900 validation pairs, and ~1,000-1,300 test pairs.
- IWSLT-17: Used for supervised and zero-shot experiments. We report results for Dutch (`Nl`), English (`En`), German (`De`), Italian (`It`), and Romanian (`Ro`). This dataset contains ~220,000 training sentence pairs (for all language pairs except for the zero-shot ones), ~900 validation pairs, and ~1,100 test pairs.

Data Preprocessing. We preprocess our data using a modified version of the Moses tokenizer (Koehn et al., 2007) that correctly handles escaped HTML characters. We also perform some Unicode character normalization and cleaning. While training, we only consider sentences up to length 50. For both datasets, we generate a per-language vocabulary consisting of the most frequently occurring words, while ignoring words that appear less than 5 times in the whole corpus, and capping the vocabulary size to 20,000 words.

Results. Our results for the IWSLT-15 experiments are shown in Table 1. It is clear that our approach consistently outperforms both the corresponding pairwise model and GML. Furthermore, its advantage grows larger in the low-resource setting (up to 5.06 BLEU score difference, or a 2.4× increase), which is expected due to the extensive parameter sharing in our model. For this dataset, there exist some additional published state-of-the-art results not shown in Tables 1 and 2. Huang et al. (2018) report a BLEU score of 28.07 for the `En→Vi` task, while our model is able to achieve a score of **29.03**. Furthermore, Ha et al. (2016) report a BLEU score of 25.87 for the

[5] We use our own implementation of `GML` in order to obtain a fair comparison, in terms of the whole MT pipeline. We have modified it to use the same per-language vocabularies that we use for our approaches, as the proposed shared BPE vocabulary fails to perform well for the considered datasets.

En→De task, while our model is able to achiev a score of **26.77**.[6] Our results for the IWSLT-1 experiments are shown in Table 2.[7] Again, ou method consistently outperforms both PNMT ar GML, in both the supervised and the zero-shot se tings. Furthermore, the results indicate that ou model performance is robust to different sizes o the language embeddings and the choice of M' fo controllable parameter sharing. It only underpe forms in the degenerate case where $M' = 1$. It also worth noting that, in the fully supervised setting, GML, the current state-of-the-art in the multilingual setting, underperforms the pairwise models.

The presented results provide evidence that our proposed approach is able to significantly improve performance, without requiring extensive tuning.

Language Embeddings. An important aspect of our model is that it learns language embeddings. In Figure 2 we show pairwise cosine distances between the learned language embeddings for our fully supervised experiments. There are some interesting patterns that indicate that the learned language embeddings are reasonable. For example, we observe that German (De) and Dutch (Nl) are most similar for the IWSLT-17 dataset, with Italian (It) and Romanian (Ro) coming second. Furthermore, Romanian and German are the furthest apart for that dataset. These relationships agree with linguistic knowledge about these languages and the families they belong to. We see similar patterns in the IWSLT-15 results but we focus on IWSLT-17 here, because it is a larger, better quality, dataset with more supervised language pairs. These results are encouraging for analyzing such embeddings to discover relationships between languages that were previously unknown. For example, perhaps surprisingly, French (Fr) and Vietnamese (Vi) appear to be significantly related for the IWSLT-15 dataset results. This is likely due to French influence in Vietnamese because to the occupation of Vietnam by France during the 19th and 20th centuries (Marr, 1981).

[6]We were unable to find reported state-of-the-art results for the rest of the language pairs.

[7]Note that, our results for IWSLT-17 are not comparable to those of the official challenge report (Cettolo et al., 2017), as we use less training data, a smaller baseline model, and our evaluation pipeline potentially differs. However, the numbers presented for all methods in this paper are comparable, as they were all obtained using the same baseline model and evaluation pipeline.

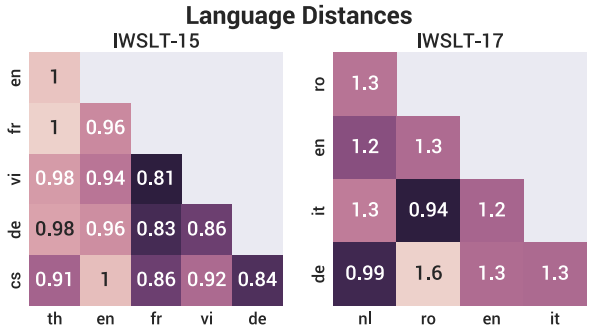

Figure 2: Pairwise cosine distance for all language pairs in the IWSLT-15 and IWSLT-17 datasets. **Darker** colors represent more similar languages.

4.1 Implementation and Reproducibility

Along with this paper we are releasing an implementation of our approach and experiments as part of a new Scala framework for machine translation.[8] It is built on top of TensorFlow Scala (Platanios, 2018) and follows the modular NMT design (described in Section 2.1) that supports various NMT models, including our baselines (e.g., Johnson et al. (2017)). It also contains data loading and preprocessing pipelines that support multiple datasets and languages, and is more efficient than other packages (e.g., `tf-nmt`[9]). Furthermore, the framework supports various vocabularies, among which we provide a new implementation for the byte-pair encoding (BPE) algorithm (Sennrich et al., 2016b) that is 2 to 3 orders of magnitude faster than the released one.[10] All experiments presented in this paper were performed using version `0.1.0` of the framework.

5 Related Work

Interlingual translation (Richens, 1958) has been the object of many research efforts. For a long time, before the move to NMT, most practical machine translation systems only focused on individual language pairs. Since the success of end-to-end NMT approaches such as the encoder-decoder framework (Sutskever et al., 2014; Bahdanau et al., 2015; Cho et al., 2014), recent work has tried to extend the framework to multi-lingual translation. An early approach was Dong et al. (2015) who performed one-to-many translation with a separate attention mechanism for each decoder. Luong et al. (2016) extended this idea with a focus on multi-task learning and multiple encoders and decoders, operating in a single shared vector space. The same architecture is used in

[8]https://github.com/eaplatanios/symphony-mt

[9]https://github.com/tensorflow/nmt

[10]https://github.com/rsennrich/subword-nmt

(Caglayan et al., 2016) for translation across multiple modalities. Zoph and Knight (2016) flipped this idea with a many-to-one translation model, however requiring the presence of a multi-way parallel corpus between all the languages, which is difficult to obtain. Lee et al. (2017) used a single character-level encoder across multiple languages by training a model on a many-to-one translation task. Closest to our work are more recent approaches, already described in Section 2 (Firat et al., 2016a; Johnson et al., 2017; Ha et al., 2016), that attempt to enforce different kinds of parameter sharing across languages.

Parameter sharing in multilingual NMT naturally enables semi-supervised and zero-shot learning. Unsupervised learning has been previously explored with key ideas such as back-translation (Sennrich et al., 2016a), dual learning (He et al., 2016), common latent space learning (Lample et al., 2018), etc. In the vein of multilingual NMT, Artetxe et al. (2018) proposed a model that uses a shared encoder and multiple decoders with a focus on unsupervised translation. The entire system uses cross-lingual embeddings and is trained to reconstruct its input using only monolingual data. Zero-shot translation was first attempted in (Firat et al., 2016b) who performed zero-zhot translation using their pre-trained multi-way multilingual model, fine-tuning it with pseudo-parallel data generated by the model itself. This was recently extended using a teacher-student framework (Chen et al., 2017). Later, zero-shot translation without any additional steps was attempted in (Johnson et al., 2017) using their shared encoder-decoder network. An iterative training procedure that leverages the duality of translations directly generated by the system for zero-shot learning was proposed by Lakew et al. (2017). For extremely low resource languages, Gu et al. (2018) proposed sharing lexical and sentence-level representations across multiple source languages with a single target language. Closely related is the work of Cheng et al. (2016) who proposed the joint training of source-to-pivot and pivot-to-target NMT models.

Ha et al. (2018) are probably the first to introduce a similar idea to that of having one network (called a *hypernetwork*) generate the parameters of another. However, in that work, the input to the hypernetwork are structural features of the original network (e.g., layer size and index). Al-Shedivat et al. (2017) also propose a related method where a neural network generates the parameters of a linear model. Their focus is mostly on interpretability (i.e., knowing which features the network considers important). However, to our knowledge, there is no previous work which proposes having a network generate the parameters of another deep neural network (e.g., a recurrent neural network), using some well-defined context based on the input data. This context, in our case, is the language of the input sentences to the translation model, along with the target translation language.

6 Conclusion and Future Directions

We have presented here a novel *contextual parameter generation* approach to neural machine translation. Our resulting system, which outperforms other state-of-the-art systems, uses a standard pairwise encoder-decoder architecture. However, it differs from earlier approaches by incorporating a component that generates the parameters to be used by the encoder and the decoder for the current sentence, based on the source and target languages, respectively. We refer to this novel component as the *contextual parameter generator*. The benefit of this approach is that it dramatically improves the ratio of the number of parameters to be learned, to the number of training examples available, by leveraging shared structure across different languages. Thus, our approach does not require any extra machinery such as back-translation, dual learning, pivoting, or multilingual word embeddings. It rather relies on the simple idea of *treating language as a context within which to encode/decode*. We also showed that the proposed approach is able to achieve state-of-the-art performance without requiring any tuning. Finally, we performed a basic analysis of the learned language embeddings, which showed that cosine distances between the learned language embeddings reflect well known similarities among language pairs such as German and Dutch.

In the future, we want to extend the concept of the contextual parameter generator to more general settings, such as translating between different modalities of data (e.g., image captioning). Furthermore, based on the discussion of Section 3.3, we hope to develop an adaptable, never-ending learning (Mitchell et al., 2018) NMT system.

Acknowledgments

We would like to thank Otilia Stretcu, Abulhair Saparov, and Maruan Al-Shedivat for the useful feedback they provided in early versions of this paper. This research was supported in part by AFOSR under grant FA95501710218.

References

Maruan Al-Shedivat, Avinava Dubey, and Eric P Xing. 2017. Contextual Explanation Networks. *CoRR*, abs/1705.10301.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised Neural Machine Translation. In *International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations*.

Ozan Caglayan, Walid Aransa, Yaxing Wang, Marc Masana, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, and Joost van de Weijer. 2016. Does Multimodality Help Human and Machine for Translation and Image Captioning? In *Proceedings of the First Conference on Machine Translation*, volume 2.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichi Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 Evaluation Campaign. In *Proceedings of the 14th International Workshop on Spoken Language Translation*.

Yun Chen, Yang Liu, Yong Cheng, and Victor O K Li. 2017. A Teacher-Student Framework for Zero-Resource Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1925–1935.

Yong Cheng, Yang Liu, Qian Yang, Maosong Sun, and Wei Xu. 2016. Neural Machine Translation with Pivot Languages. *CoRR*, abs/1611.04928.

Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

Josep Maria Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. SYSTRAN's Pure Neural Machine Translation Systems. *CoRR*, abs/1610.05540.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In *Proceedings of NAACL-HLT*, pages 866–875.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman-Vural, and Kyunghyun Cho. 2016b. Zero-Resource Translation with Multi-Lingual Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O K Li. 2018. Universal Neural Machine Translation for Extremely Low Resource Languages. In *Proceedings of NAACL-HLT*, pages 344–354.

David Ha, Andrew Dai, and Quoc V Le. 2018. HyperNetworks. In *International Conference on Learning Representations*.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. In *Proceedings of the 13th International Workshop on Spoken Language Translation*.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. 2016. Dual Learning for Machine Translation. In *Advances in Neural Information Processing Systems*, pages 820–828.

Po-Sen Huang, Chong Wang, Sitao Huang, Dengyong Zhou, and Li Deng. 2018. Towards Neural Phrase-Based Machine Translation. In *International Conference on Learning Representations*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Googles Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. In *Transactions of the Association for Computational Linguistics*, volume 5, pages 339–351.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180.

Surafel M Lakew, Quintino F Lotito, Negri Matteo, Turchi Marco, and Federico Marcello. 2017. Improving Zero-Shot Translation of Low-Resource

Languages. In *Proceedings of the 14th International Workshop on Spoken Language Translation*, pages 113–119.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised Machine Translation Using Monolingual Corpora Only. In *International Conference on Learning Representations*.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully Character-Level Neural Machine Translation without Explicit Segmentation. 5:365–378.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task Sequence to Sequence Learning. In *International Conference on Learning Representations*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

David G. Marr. 1981. Language and Literacy. In *Vietnamese Tradition on Trial, 1920-1945*, pages 136–189. University of California Press.

Paul Michel and Graham Neubig. 2018. Extreme Adaptation for Personalized Neural Machine Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 312–318.

Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bishan Yang, Justin Betteridge, Andrew Carlson, Bhavana Dalvi, Matt Gardner, Bryan Kisiel, Jayant Krishnamurthy, Ni Lao, Kathryn Mazaitis, Thahir P. Mohamed, Ndapa Nakashole, Emmanouil A. Platanios, Alan Ritter, Mehdi Samadi, Burr Settles, Richard Wang, Derry Wijaya, Abhinav Gupta, Xinlei Chen, Abulhair Saparov, Malcolm Greaves, and Joel Welling. 2018. Never-Ending Learning. *Communications of the ACM*, 61(5):103–115.

Emmanouil A. Platanios. 2018. TensorFlow Scala. `https://github.com/eaplatanios/tensorflow_scala`.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the Convergence of Adam and Beyond. In *International Conference on Learning Representations*.

Richard H Richens. 1958. Interlingual Machine Translation. *The Computer Journal*, 1(3):144–147.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, abs/1609.08144.

Barret Zoph and Kevin Knight. 2016. Multi-Source Neural Translation. In *Proceedings of NAACL-HLT*, pages 30–34.