Meta-Learning for Low-Resource Neural Machine Translation

Anonymous EMNLP submission

Abstract

In this paper, we propose to extend the recently introduced model-agnostic meta-learning algorithm (MAML, Finn et al., 2017) for low-resource neural machine translation (NMT). We frame low-resource translation as a meta-learning problem, and we learn to adapt to low-resource languages based on multilingual high-resource language tasks. We use the universal lexical representation (Gu et al., 2018b) to overcome the input-output mismatch across different languages. We evaluate the proposed meta-learning strategy using eighteen European languages (Bg, Cs, Da, De, El, Es, Et, Fr, Hu, It, Lt, Ni, Pl, Pt, Sk, Sl, Sv and Ru) as source tasks and five diverse languages (Ro, Lv, Fi, Tr and Ko) as target tasks. We show that the proposed approach significantly outperforms the multilingual, transfer learning based approach (Zoph et al., 2016) and enables us to train a competitive NMT system with only a fraction of training examples. For instance, the proposed approach can achieve as high as 22.04 BLEU on Romanian-English WMT’16 by seeing only 16,000 translated words (∼600 parallel sentences).

1 Introduction

Despite the massive success brought by neural machine translation (NMT, Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017), it has been noticed that the vanilla NMT often lags behind conventional machine translation systems, such as statistical phrase-based translation systems (PBMT, Koehn et al., 2003), for low-resource language pairs (see, e.g., Koehn and Knowles, 2017). In the past few years, various approaches have been proposed to address this issue. The first attempts at tackling this problem exploited the availability of monolingual corpora (Gulcehre et al., 2015; Sennrich et al., 2015; Zhang and Zong, 2016). It was later followed by approaches based on multilingual translation, in which the goal was to exploit knowledge from high-resource language pairs by training a single NMT system on a mix of high-resource and low-resource language pairs (Firat et al., 2016a,b; Lee et al., 2016; Johnson et al., 2016; Ha et al., 2016b). Its variant, transfer learning, was also proposed by Zoph et al. (2016), in which an NMT system is pretrained on a high-resource language pair before being fine-tuned on a target low-resource language pair.

In this paper, we follow up on these latest approaches based on multilingual NMT and propose a meta-learning algorithm for low-resource neural machine translation. We start by arguing that the recently proposed model-agnostic meta-learning algorithm (MAML, Finn et al., 2017) could be applied to low-resource machine translation by viewing language pairs as separate tasks. This view enables us to use MAML to find the initialization of model parameters that facilitate fast adaptation for a new language pair with a minimal amount of training examples. For instance, the proposed approach can achieve as high as 22.04 BLEU on Romanian-English WMT’16 by seeing only 16,000 translated words (∼600 parallel sentences).
decreases.

2 Background

Neural Machine Translation (NMT) Given a source sentence \( X = \{x_1, \ldots, x_T\} \), a neural machine translation model factors the distribution over possible output sentences \( Y = \{y_1, \ldots, y_T\} \) into a chain of conditional probabilities with a left-to-right causal structure:

\[
p(Y|X; \theta) = \prod_{t=1}^{T+1} p(y_t|y_{0:t-1}, x_{1:T}; \theta),
\]

where special tokens \( y_0 (\langle \text{bos}\rangle) \) and \( y_{T+1} (\langle \text{eos}\rangle) \) are used to represent the beginning and the end of a target sentence. These conditional probabilities are parameterized using a neural network. Typically, an encoder-decoder architecture (Sutskever et al. 2014; Cho et al. 2014; Bahdanau et al. 2015) with a RNN-based decoder is used. More recently, architectures without any recurrent structures (Gehring et al. 2017; Vaswani et al. 2017) have been proposed and shown to speedup training while achieving state-of-the-art performance.

Low Resource Translation NMT is known to easily over-fit and result in an inferior performance when the training data is limited (Koehn and Knowles 2017). In general, there are two ways for handling the problem of low resource translation: (1) utilizing the resource of unlabeled monolingual data, and (2) sharing the knowledge between low- and high-resource language pairs. Many research efforts have been spent on incorporating the monolingual corpora into machine translation, such as multi-task learning (Gulcehre et al. 2015; Zhang and Zong 2016), back-translation (Sennrich et al. 2015), dual learning (He et al. 2016) and unsupervised machine translation with monolingual corpora only for both sides (Artetxe et al. 2017b; Lample et al. 2017; Yang et al. 2018).

For the second approach, prior researches have worked on methods to exploit the knowledge of auxiliary translations, or even auxiliary tasks. For instance, Cheng et al. (2016); Chen et al. (2017); Lee et al. (2017); Chen et al. (2018) investigate the use of a pivot to build a translation path between two languages even without any directed resource. The pivot can be a third language or even an image in multimodal domains. When pivots are not easy to obtain, Firat et al. (2016a); Lee et al. (2016); Johnson et al. (2016) have shown that the structure of NMT is suitable for multilingual machine translation. Gu et al. (2018a) also showed that such a multilingual NMT system could improve the performance of low resource translation by using a universal lexical representation to share embedding information across languages. All the previous work for multilingual NMT assume the joint training of multiple high-resource languages naturally results in a universal space (for both the input representation and the model) which, however, is not necessarily true, especially for very low resource cases.

Meta Learning In the machine learning community, meta-learning, or learning-to-learn, has recently received interests. Meta-learning tries to solve the problem of “fast adaptation on new training data.” One of the most successful applications of meta-learning has been on few-shot (or one-shot) learning (Lake et al. 2015), where a neural network is trained to readily learn to classify inputs based on only one or a few training examples. There are two categories of meta-learning:

1. learning a meta-policy for updating model parameters (see, e.g., Andrychowicz et al. 2016; Ha et al. 2016a; Mishra et al. 2017).
2. learning a good parameter initialization for fast adaptation (see, e.g., Finn et al. 2017; Vinyals et al. 2016; Snell et al. 2017).

In this paper, we propose to use a meta-learning algorithm for low-resource neural machine translation based on the second category. More specifically, we extend the idea of model-agnostic meta-learning (MAML, Finn et al. 2017) in the multilingual scenario.

3 Meta Learning for Low-Resource Neural Machine Translation

The underlying idea of MAML is to use a set of source tasks \( \{T^1, \ldots, T^K\} \) to find the initialization of parameters \( \theta^i \) from which learning a target task \( T^0 \) would require only a small number of training examples. In the context of machine translation, this amounts to using many high-resource language pairs to find good initial parameters and training a new translation model on a low-resource language starting from the found initial parameters. This process can be understood as

\[
\theta^i = \text{Learn}(T^0; \text{MetaLearn}(T^1, \ldots, T^K)).
\]
That is, we *meta-learn* the initialization from auxiliary tasks and continue to *learn* the target task. We refer the proposed meta-learning method for NMT to MetaNMT. See Fig. 1 for the overall illustration.

### 3.1 Learn: language-specific learning

Given any initial parameters $\theta^0$ (which can be either random or meta-learned), the prior distribution of the parameters of a desired NMT model can be defined as an isotropic Gaussian:

$$
\theta_i \sim N(\theta_i^0, 1/\beta),
$$

where $1/\beta$ is a variance. With this prior distribution, we formulate the language-specific learning process Learn$(D_T; \theta^0)$ as maximizing the log-posterior of the model parameters given data $D_T$:

$$
\text{Learn}(D_T; \theta^0) = \arg \max_{\theta} \mathcal{L}^D_T(\theta)
$$

$$
= \arg \max_{\theta} \sum_{(X,Y) \in D_T} \log p(Y|X, \theta) - \beta \| \theta - \theta^0 \|^2,
$$

where we assume $p(X|\theta)$ to be uniform. The first term above corresponds to the maximum likelihood criterion often used for training a usual NMT system. The second term discourages the newly learned model from deviating too much from the initial parameters, alleviating the issue of over-fitting when there is not enough training data. In practice, we solve the problem above by maximizing the first term with gradient-based optimization and early-stopping after only a few update steps. Thus, in the low-resource scenario, finding a good initialization $\theta^0$ strongly correlates the final performance of the resulting model.

### 3.2 MetaLearn

We find the initialization $\theta^0$ by repeatedly simulating low-resource translation scenarios using auxiliary, high-resource language pairs. Following Finn et al. (2017) we achieve this goal by defining the meta-objective function as

$$
\mathcal{L}(\theta) = \mathbb{E}_k \mathbb{E}_{D_T, D_T'} \left[ \sum_{(X,Y) \in D_T'} \log p(Y|X; \text{Learn}(D_T'; \theta)) \right],
$$

where $k \sim \mathcal{U}\{\{1, \ldots, K\}\}$ refers to one meta-learning episode, and $D_T, D_T'$ follow the uniform distribution over $T$’s data.

We maximize the meta-objective function using stochastic approximation (Robbins and Monro, 1951) with gradient descent. For each episode, we uniformly sample one source task at random, $T^k$. We then sample two subsets of training examples independently from the chosen task, $D_T^k$ and $D_T'^k$. We use the former to *simulate* language-specific learning and the latter to *evaluate* its outcome. Assuming a single gradient step is taken only the with learning rate $\eta$, the simulation is:

$$
\theta_k' = \text{Learn}(D_T^k; \theta) = \theta - \eta \nabla_{\theta} \mathcal{L}^{D_T^k}(\theta).
$$

Once the simulation of learning is done, we evaluate the updated parameters $\theta_k'$ on $D_T'^k$. The gradient computed from this evaluation, which we refer to as *meta-gradient*, is used to update the meta model $\theta$. It is possible to aggregate multiple episodes of source tasks before updating $\theta$:

$$
\theta \leftarrow \theta - \eta' \sum_k \nabla_{\theta} \mathcal{L}^{D_T'^k}(\theta_k'),
$$
where $\eta'$ is the meta learning rate.

Unlike a usual learning scenario, the resulting model $\theta^0$ from this meta-learning procedure is not necessarily a good model on its own. It is however a good starting point for training a good model using only a few steps of learning. In the context of machine translation, this procedure can be understood as finding the initialization of a neural machine translation system that could quickly adapt to a new language pair by simulating such a fast adaptation scenario using many high-resource language pairs.

**Meta-Gradient** We use the following approximation property

$$H(x) \approx \nabla(x + \nu v) - \nabla(x)$$

to approximate the meta-gradient:

$$\nabla_{\theta} \mathcal{L}^{D'}(\theta') = \nabla_{\theta} \mathcal{L}^{D'}(\theta') \nabla_{\theta} (\theta - \eta \nabla_{\theta} \mathcal{L}^{D} (\theta))$$

$$= \nabla_{\theta} \mathcal{L}^{D'}(\theta') - \eta \nabla_{\theta} \mathcal{L}^{D'}(\theta') H_{\Theta}(\mathcal{L}^{D}(\theta))$$

$$\approx \nabla_{\theta} \mathcal{L}^{D'}(\theta') - \frac{\eta}{\nu} \left[ \nabla_{\theta} \mathcal{L}^{D}(\theta) \right]$$

where $\nu$ is a small constant and

$$\hat{\theta} = \theta + \nu \nabla_{\theta} \mathcal{L}^{D'}(\theta').$$

In practice, we find that it is also possible to ignore the second-order term, ending up with the following simplified update rule:

$$\nabla_{\theta} \mathcal{L}^{D'}(\theta') \approx \nabla_{\theta} \mathcal{L}^{D'}(\theta').$$

(3)

**Related Work: Multilingual Transfer Learning**

The proposed MetaNMT differs from the existing framework of multilingual translation (Lee et al., 2016; Johnson et al., 2016; Gu et al., 2018b) or transfer learning (Zoph et al., 2016). The latter can be thought of as solving the following problem:

$$\max_{\theta} \mathcal{L}^{\text{multi}}(\theta) = \mathbb{E}_{k} \left[ \sum_{(X,Y) \in D_k} \log p(Y|X; \theta) \right],$$

where $D_k$ is the training set of the $k$-th task, or language pair. The target low-resource language pair could either be a part of joint training or be trained separately starting from the solution $\theta^0$ found from solving the above problem.

The major difference between the proposed MetaNMT and these multilingual transfer approaches is that the latter do not consider how learning happens with the target, low-resource language pair. The former explicitly incorporates the learning process within the framework by simulating it repeatedly in Eq. (2). As we will see later in the experiments, this results in a substantial gap in the final performance on the low-resource task.

**Illustration** In Fig. 2, we contrast transfer learning, multilingual learning and meta-learning using three source language pairs (Fr-En, Es-En and Pt-En) and two target pairs (Ro-En and Lv-En). Transfer learning trains an NMT system specifically for a source language pair (Es-En) and finetunes the system for each target language pair (Ro-En, Lv-En). Multilingual learning often trains a single NMT system that can handle many different language pairs (Fr-En, Pt-En, Es-En), which may or may not include the target pairs (Ro-En, Lv-En). If not, it finetunes the system for each target pair, similarly to transfer learning. Both of these however aim at directly solving the source tasks. On the other hand, meta-learning trains the NMT system to be *useful for fine-tuning* on various tasks including the source and target tasks. This is done by repeatedly simulating the learning process on low-resource languages using many high-resource language pairs (Fr-En, Pt-En, Es-En).

### 3.3 Unified Lexical Representation

**I/O mismatch across language pairs** One major challenge that limits applying meta-learning for low resource machine translation is that the approach outlined above assumes the input and output spaces are shared across all the source and target tasks. This, however, does not apply to ma-
chine translation in general due to the vocabulary mismatch across different languages. In multilingual translation, this issue has been tackled by using a vocabulary of sub-words (Sennrich et al. 2015) or characters (Lee et al., 2016) shared across multiple languages. This surface-level sharing is however limited, as it cannot be applied to languages exhibiting distinct orthography (e.g., Indo-European languages vs. Korean.)

**Universal Lexical Representation (ULR)** We tackle this issue by dynamically building a vocabulary specific to each language using a key-value memory network (Miller et al. 2016; Gulcehre et al. 2018), as was done successfully for low-resource machine translation recently by Gu et al. (2018). We start with multilingual word embedding matrices $\epsilon^k_{query} \in \mathbb{R}^{V_k \times d}$ pretrained on large monolingual corpora, where $V_k$ is the vocabulary of the $k$-th language. These embedding vectors can be obtained with small dictionaries of seed word pairs (Artetxe et al. 2017a; Smith et al. 2017) or in a fully unsupervised manner (Zhang et al. 2017; Alexis et al. 2018). We take one of these languages $k$ to build universal lexical representation consisting of a universal embedding matrix $\epsilon_u \in \mathbb{R}^{M \times d}$ and a corresponding key matrix $\epsilon_{key} \in \mathbb{R}^{M \times d}$, where $M < |V'_k|$. Both $\epsilon^k_{query}$ and $\epsilon_{key}$ are fixed during meta-learning. We then compute the language-specific embedding of token $x$ from the language $k$ as the convex sum of the universal embedding vectors by

$$e^0[x] = \sum_{i=1}^{M} \alpha_i \epsilon_u[i],$$

where $\alpha_i \propto \exp \left\{ -\frac{1}{\tau} \epsilon_{key}[i]^{\top} A_{\epsilon^k_{query}}[x] \right\}$ and $\tau$ is set to 0.05. This approach allows us to handle languages with different vocabularies using a fixed number of shared parameters ($\epsilon_u$, $\epsilon_{key}$ and $A$.)

**Learning of ULR** It is not desirable to update the universal embedding matrix $\epsilon_u$ when fine-tuning on a small corpus which contains a limited set of unique tokens in the target language, as it could adversely influence the other tokens’ embedding vectors. We thus estimate the change to each embedding vector induced by language-specific learning by a separate parameter $\Delta \epsilon^k[x]$:

$$\epsilon^k[x] = e^0[x] + \Delta \epsilon^k[x].$$

During language-specific learning, the ULR $e^0[x]$ is held constant, while only $\Delta \epsilon^k[x]$ is updated, starting from an all-zero vector. On the other hand, we hold $\Delta \epsilon^k[x]$’s constant while updating $\epsilon_u$ and $A$ during the meta-learning stage.

4 Experimental Settings

4.1 Dataset

**Target Tasks** We show the effectiveness of the proposed meta-learning method for low resource NMT with extremely limited training examples on five diverse target languages: Romanian (Ro) from WMT’16, Latvian (Lv), Finnish (Fi), Turkish (Tr) from WMT’17 and Korean (Ko) from Korean Parallel Dataset. We use the officially provided train, dev and test splits for all these languages. The statistics of these languages are presented in Table 1. We simulate the low-resource translation scenarios by randomly sub-sampling the training set with different sizes.

**Source Tasks** We use the following languages from Europarl\textsuperscript{3} Bulgarian (Bg), Czech (Cs), Danish (Da), German (De), Greek (El), Spanish (Es), Estonian (Et), French (Fr), Hungarian (Hu), Italian (It), Lithuanian (Lt), Dutch (Nl), Polish (Pl), Portuguese (Pt), Slovak (Sk), Slovene (Sl) and Swedish (Sv), in addition to Russian (Ru)\textsuperscript{4} to learn the initialization for fine-tuning. In our experiments, different combinations of source tasks are explored to see the effects from the source tasks.

**Validation** We pick either Ro-En or Lv-En as a validation set for meta-learning and test the generalization capability on the remaining target tasks. This allows us to study the strict form of meta-learning, in which target tasks are unknown during training.

<table>
<thead>
<tr>
<th># of sents.</th>
<th># of En tokens</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ro-En</td>
<td>0.61 M</td>
<td>16.66 M</td>
<td>−</td>
</tr>
<tr>
<td>Lv-En</td>
<td>4.46 M</td>
<td>67.24 M</td>
<td>20.24</td>
</tr>
<tr>
<td>Fi-En</td>
<td>2.63 M</td>
<td>64.50 M</td>
<td>17.38</td>
</tr>
<tr>
<td>Tr-En</td>
<td>0.21 M</td>
<td>5.58 M</td>
<td>15.45</td>
</tr>
<tr>
<td>Ko-En</td>
<td>0.09 M</td>
<td>2.33 M</td>
<td>6.88</td>
</tr>
</tbody>
</table>

Table 1: Statistics of full datasets of the target language pairs. BLEU scores on the dev and test sets are reported from a supervised Transformer model with the same architecture.

\textsuperscript{3}http://www.statmt.org/wmt16/translation-task.html
\textsuperscript{4}http://www.statmt.org/wmt17/translation-task.html
\textsuperscript{5}https://sites.google.com/site/koreanparalleldata/
\textsuperscript{6}http://www.statmt.org/europarl/
both training and model selection.

Preprocessing and ULR Initialization  As described in §3.3, we initialize the query embedding vectors $\epsilon_{query}$ of all the languages. For each language, we use the monolingual corpora built from Wikipedia and the parallel corpus. The concatenated corpus is first tokenized and segmented using byte-pair encoding (BPE, Sennrich et al. 2016), resulting in 40,000 subwords for each language. We then estimate word vectors using fastText (Bojanowski et al., 2016), resulting in $40,000$, using byte-pair encoding (BPE, Sennrich et al., 2016), resulting in $40,000$, resulting in $40,000$. We use the multilingual word vectors of the 20,000 most frequent words in English to form the universal embedding matrix $\epsilon_u$.

4.2 Model and Learning

Model  We utilize the recently proposed Transformer (Vaswani et al. 2017) as an underlying NMT system. We implement Transformer in this paper based on (Gu et al., 2018a) and modify it to use the universal lexical representation from §3.3. We use the default set of hyperparameters ($d_{model} = d_{hidden} = 512$, $n_{layer} = 6$, $n_{head} = 8$, $n_{batch} = 4000$, $t_{warmup} = 16000$) for all the language pairs and across all the experimental settings. We refer the readers to (Vaswani et al., 2017; Gu et al., 2018a) for the details of the model. However, since the proposed meta-learning method is model-agnostic, it can be easily extended to any other NMT architectures, e.g. RNN-based sequence-to-sequence models with attention (Bahdanau et al., 2015).

Learning  We meta-learn using various sets of source languages to investigate the effect of source task choice. For each episode, by default, we use a single gradient step of language-specific learning with Adam (Kingma and Ba, 2014) per computing the meta-gradient, which is computed by the first-order approximation in Eq. (3).

For each target task, we sample training examples to form a low-resource task. We build tasks of 4k, 16k, 40k and 160k English tokens for each language. We randomly sample the training set five times for each experiment and report the average score and its standard deviation. Each fine-tuning
Table 2: BLEU Scores w.r.t. the source task set for all five target tasks.

<table>
<thead>
<tr>
<th>Meta-Train</th>
<th>Ro-En</th>
<th>Lv-En</th>
<th>Fi-En</th>
<th>Tr-En</th>
<th>Ko-En</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>zero</td>
<td>zero</td>
<td>zero</td>
<td>zero</td>
<td>zero</td>
</tr>
<tr>
<td>All</td>
<td>0.00 ± 0.00</td>
<td>0.00 ± 0.00</td>
<td>0.00 ± 0.00</td>
<td>0.00 ± 0.00</td>
<td>0.00 ± 0.00</td>
</tr>
<tr>
<td>Es</td>
<td>9.20 ± 15.71 ± 0.22</td>
<td>2.23 ± 4.65 ± 0.12</td>
<td>7.05 ± 5.55 ± 0.08</td>
<td>0.16 ± 5.14 ± 0.03</td>
<td>0.63 ± 5.14 ± 0.09</td>
</tr>
<tr>
<td>Es Fr It Pt</td>
<td>12.35 ± 17.46 ± 0.41</td>
<td>2.86 ± 5.05 ± 0.04</td>
<td>3.71 ± 6.08 ± 0.01</td>
<td>2.17 ± 5.46 ± 0.20</td>
<td>0.31 ± 1.70 ± 0.14</td>
</tr>
<tr>
<td>Es Fr It Pt De Ru</td>
<td>13.88 ± 18.54 ± 0.19</td>
<td>3.88 ± 5.63 ± 0.11</td>
<td>4.93 ± 6.80 ± 0.04</td>
<td>2.49 ± 5.82 ± 0.20</td>
<td>0.82 ± 1.80 ± 0.07</td>
</tr>
<tr>
<td>Es Fr It Pt De Ru</td>
<td>15.93 ± 20.00 ± 0.27</td>
<td>5.15 ± 7.19 ± 0.17</td>
<td>6.62 ± 7.58 ± 0.22</td>
<td>3.20 ± 5.02 ± 0.11</td>
<td>1.19 ± 2.16 ± 0.09</td>
</tr>
<tr>
<td>All</td>
<td>18.12 ± 22.04 ± 0.23</td>
<td>9.58 ± 10.44 ± 0.17</td>
<td>11.39 ± 12.63 ± 0.22</td>
<td>5.34 ± 8.97 ± 0.08</td>
<td>1.96 ± 3.97 ± 0.10</td>
</tr>
</tbody>
</table>

Figure 4: BLEU Scores w.r.t. the size of the target task’s training set.

is done on a training set, early-stopped on a validation set and evaluated on a test set.

Fine-tuning Strategies The transformer consists of three modules; embedding, encoder and decoder. We update all three modules during meta-learning, but during fine-tuning, we can selectively tune only a subset of these modules. Following (Zoph et al., 2016), we consider three fine-tuning strategies; (1) fine-tuning all the modules (all), (2) fine-tuning the embedding and encoder, but freezing the parameters of the decoder (emb+enc) and (3) fine-tuning the embedding only (emb).

5 Results

vs. Multilingual Transfer Learning We meta-learn the initial models on all the source tasks using either Ro-En or Lv-En as a validation task. We also train the initial models to be multilingual translation systems. We fine-tune them using the four target tasks (Ro-En, Lv-En, Fi-En and Tr-En; 16k tokens each) and compare the proposed meta-learning strategy and the multilingual, transfer learning strategy. As presented in Fig. [3], the proposed learning approach significantly outperforms the multilingual, transfer learning strategy across all the target tasks regardless of which target task was used for early stopping. We also notice that the emb+enc strategy is most effective for both meta-learning and transfer learning approaches. With the proposed meta-learning and emb+enc fine-tuning, the final NMT systems trained using only a fraction of all available training examples achieve 2/3 (Ro-En) and 1/2 (Lv-En, Fi-En and Tr-En) of the BLEU score achieved by the models trained with full training sets.

Impact of Validation Tasks Similarly to training any other neural network, meta-learning still requires early-stopping to avoid overfitting to a specific set of source tasks. In doing so, we observe that the choice of a validation task has non-negligible impact on the final performance. For instance, as shown in Fig. [3] Fi-En benefits more when Ro-En is used for validation, while the opposite happens with Tr-En. The relationship between the task similarity and the impact of a validation task must be investigated further in the future.

Training Set Size We vary the size of the target task’s training set and compare the proposed meta-learning strategy and multilingual, transfer learning strategy. We use the emb+enc fine-tuning on Ro-En and Fi-En. Fig. [4] demonstrates that the meta-learning approach is more robust to the drop in the size of the target task’s training set. The gap between the meta-learning and transfer learning grows as the size shrinks, confirming the effectiveness of the proposed approach on extremely low-resource language pairs.

Impact of Source Tasks In Table 2 we present the results on all five target tasks obtained while varying the source task set. We first see that it is
Source (Ko) | 이반에 제포되어 기소된 사람들 중에는 희석한 군 고위관리, 언론인, 정치인, 경제인 등이 포함됐다
---|---
Target | among the suspects are retired military officials, journalists, politicians and economists.
Meta-0 | MetaNMT Fine-tune
Meta-16k | google has launched a campaign to collect $11 million for refugees.

Table 3: Sample translations for Tr-En and Ko-En highlight the impact of fine-tuning which results in syntactically better formed translations. We highlight tokens of interest in terms of reordering.

Inspecting these examples provides the insight into the proposed meta-learning algorithm. For instance, we observe that the meta-learned model without any fine-tuning produces a word-by-word translation in the first example (Tr-En), which is due to the successful use of the universal lexical representation and the meta-learned initialization. The system however cannot reorder tokens from Turkish to English, as it has not seen any training example of Tr-En. After seeing around 600 sentence pairs (16K English tokens), the model rapidly learns to correctly reorder tokens to form a better translation. A similar phenomenon is observed in the Ko-En example. These cases could be found across different language pairs.

6 Conclusion

In this paper, we proposed a meta-learning algorithm for low-resource neural machine translation that exploits the availability of high-resource language pairs. We based the proposed algorithm on the recently proposed model-agnostic meta-learning and adapted it to work with multiple languages that do not share a common vocabulary using the technique of universal lexical representation, resulting in MetaNMT. Our extensive evaluation, using 18 high-resource source tasks and 5 low-resource target tasks, has shown that the proposed MetaNMT significantly outperforms the existing approach of multilingual, transfer learning in low-resource neural machine translation across all the language pairs considered.

The proposed approach opens new opportunities for neural machine translation. First, it is a principled framework for incorporating various extra sources of data, such as source- and target-side monolingual corpora. Second, it is a generic framework that can easily accommodate existing and future neural machine translation systems.
References


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


