# Importance-based Neuron Allocation for Multilingual Neural Machine Translation

**Wanying Xie**[1,2,3] **Yang Feng**[1,2*] **Shuhao Gu**[1,2] **Dong Yu**[3]
[1] Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Beijing Language and Culture University, China
xiewanying07@gmail.com, yudong@blcu.edu.cn
{fengyang, gushuhao19b}@ ict.ac.cn

## Abstract

Multilingual neural machine translation with a single model has drawn much attention due to its capability to deal with multiple languages. However, the current multilingual translation paradigm often makes the model tend to preserve the general knowledge, but ignore the language-specific knowledge. Some previous works try to solve this problem by adding various kinds of language-specific modules to the model, but they suffer from the parameter explosion problem and require specialized manual design. To solve these problems, we propose to divide the model neurons into general and language-specific parts based on their importance across languages. The general part is responsible for preserving the general knowledge and participating in the translation of all the languages, while the language-specific part is responsible for preserving the language-specific knowledge and participating in the translation of some specific languages. Experimental results on several language pairs, covering IWSLT and Europarl corpus datasets, demonstrate the effectiveness and universality of the proposed method.

## 1 Introduction

Neural machine translation(NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017) has shown its superiority and drawn much attention in recent years. Although the NMT model can achieve promising results for high-resource language pairs, it is unaffordable to train separate models for all the language pairs since there are thousands of languages in the world (Tan et al., 2019; Aharoni et al., 2019; Arivazhagan et al., 2019). A typical solution to reduce the model size and the training cost is to handle multiple languages in a single multilingual neural machine translation (MNMT) model (Ha et al., 2016; Firat et al., 2016; Johnson et al., 2017; Gu et al., 2018). The standard paradigm of MNMT proposed by Johnson et al. (2017) contains a language-shared encoder and decoder with a special language indicator in the input sentence to determine the target language.

Because different languages share all of the model parameters in the standard MNMT model, the model tends to converge to a region where there are low errors for all the languages. Therefore, the MNMT model trained on the combined data generally captures the general knowledge, but ignores the language-specific knowledge, rendering itself sub-optimal for the translation of a specific language (Sachan and Neubig, 2018; Blackwood et al., 2018; Wang et al., 2020b). To retain the language-specific knowledge, some researches turn to augment the NMT model with language-specific modules, e.g., the language-specific attention module (Blackwood et al., 2018), decoupled multilingual encoders and/or decoders (Vázquez et al., 2019; Escolano et al., 2020) and the lightweight language adapters (Bapna and Firat, 2019). However, these methods suffer from the parameter increment problem, because the number of parameters increases linearly with the number of languages. Besides, the structure, size, and location of the module have a large influence on the final performance, which requires specialized manual design. As a result, these problems often prevent the application of these methods in some scenarios.

Based on the above, we aim to propose a method that can retain the general and language-specific knowledge, and keep a stable model size as the number of language-pair increases without introducing any specialized module. To achieve this, we propose to divide the model neurons into two parts based on their importance: the general neurons

*Corresponding author: Yang Feng.

Our code can be got at https://github.com/ictnlp/NA-MNMT



*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 5725–5737
August 1–6, 2021. 

which are used to retain the general knowledge of all the languages, and the language-specific neurons which are used to retain the language-specific knowledge. Specifically, we first pre-train a standard MNMT model on all language data and then evaluate the importance of each neuron in each language pair. According to their importance, we divide the neurons into the general neurons and the language-specific neurons. After that, we fine-tune the translation model on all language pairs. In this process, only the general neurons and the corresponding language-specific neurons for the current language pair participate in training. Experimental results on different languages show that the proposed method outperforms several strong baselines.

Our contributions can be summarized as follows:

- We propose a method that can improve the translation performance of the MNMT model without introducing any specialized modules or adding new parameters.
- We show that the similar languages share some common features that can be captured by some specific neurons of the MNMT model.
- We show that some modules tend to capture the general knowledge while some modules are more essential for capturing the language-specific knowledge.

## 2 Background

In this section, we will give a brief introduction to the Transformer model (Vaswani et al., 2017) and the Multilingual translation.

### 2.1 The Transformer

We denote the input sequence of symbols as $\mathbf{x}' = (x_1, \ldots, x_J)$, the ground-truth sequence as $\mathbf{y}^* = (y_1^*, \ldots, y_{K*}^*)$ and the translation as $\mathbf{y} = (y_1, \ldots, y_K)$.

Transformer is a stacked network with $N$ identical layers containing two or three basic blocks in each layer. For a single layer in the encoder, it consists of a multi-head self-attention and a position-wise feed-forward network. For a single decoder layer, besides the above two basic blocks, a multi-head cross-attention follows multi-head self-attention. The input sequence $\mathbf{x}$ will be first converted to a sequence of vectors and fed into the encoder. Then the output of the $N$-th encoder layer will be taken as source hidden states and fed into decoder. The final output of the $N$-th decoder layer gives the target hidden states and translate the target sentences.

### 2.2 Multilingual Translation

In the standard paradigm of MNMT, all parameters are shared across languages and the model is jointly trained on multiple language pairs. We follow Johnson et al. (2017) to reuse standard bilingual NMT models for multilingual translation by altering the source input with a language token *lang*, i.e. changing $\mathbf{x}'$ to $\mathbf{x} = (lang, x_1, \ldots, x_J)$.

## 3 Approach

Our goal is to build a unified model, which can achieve good performance on all language pairs. The main idea of our method is that different neurons have different importance to the translation of different languages. Based on this, we divide them into general and language-specific ones and make general neurons participate in the translation of all the languages while language-specific neurons focus on some specific languages. Specifically, the proposed approach involves the following steps shown in Figure 1. First, we pretrain the model on the combined data of all the language pairs following the normal paradigm in Johnson et al. (2017). Second, we evaluate the importance of different neurons on these language pairs and allocate them into general neurons and language-specific neurons. Last, we fine-tune the translation model on the combined data again. It should be noted that for a specific language pair only the general neurons and the language-specific neurons for this language pair will participate in the forward and backward computation when the model is trained on this language pair. Other neurons will be zeroed out during both training and inference.

### 3.1 Importance Evaluation

The basic idea of importance evaluation is to determine which neurons are essential to all languages while which neurons are responsible for some specific languages. For a neuron $i$, its average importance $\mathcal{I}$ across language pairs is defined as follow:

$$\mathcal{I}(i) = \frac{1}{M} \sum_{m=1}^{M} \Theta^m(i), \quad (1)$$

where the $\Theta(\cdot)$ denotes the importance evaluation function and $M$ denotes the number of language

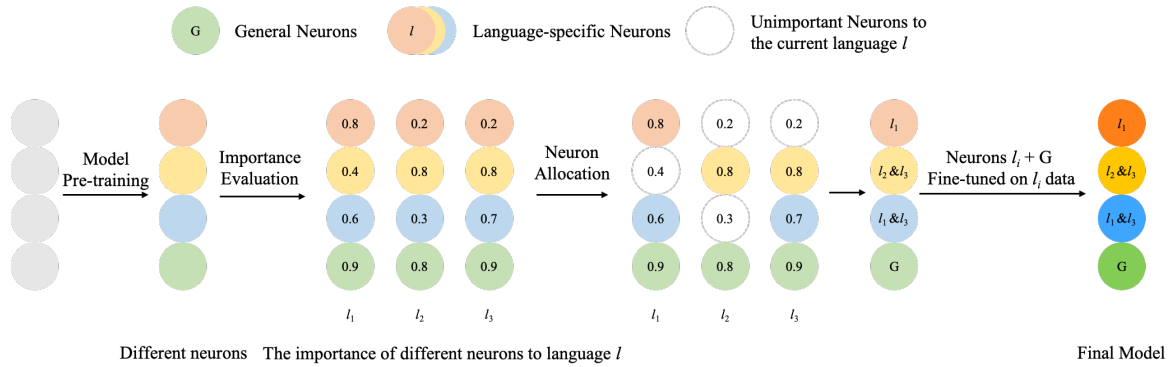



Figure 1: The whole training process of the proposed method. The red, yellow and blue circles represent language-specific neurons that are important for $l_1$, $l_2\&l_3$ and $l_1\&l_3$, respectively.

pairs. This value correlates positively with how important the neuron is to all languages. For the importance evaluation function $\Theta(\cdot)$, we adopt two schemes: one is based on the Taylor Expansion and the other is based on the Absolute Value.

**Taylor Expansion** We adopt a criterion based on the Taylor Expansion (Molchanov et al., 2017), where we directly approximate the change in loss when removing a particular neuron. Let $h_i$ be the output produced from neuron $i$ and $H$ represents the set of other neurons. Assuming the independence of each neuron in the model, the change of loss when removing a certain neuron can be represented as:

$$|\Delta\mathcal{L}(h_i)| = |\mathcal{L}(H, h_i = 0) - \mathcal{L}(H, h_i)|, \quad (2)$$

where $\mathcal{L}(H, h_i = 0)$ is the loss value if the neuron $i$ is pruned and $\mathcal{L}(H, h_i)$ is the loss if it is not pruned. For the function $\mathcal{L}(H, h_i)$, its Taylor Expansion at point $h_i = a$ is:

$$\mathcal{L}(H, h_i) = \sum_{n=0}^{N} \frac{\mathcal{L}^n(H, a)}{n!}(h_i - a)^n + R_N(h_i), \quad (3)$$

where $\mathcal{L}^n(H, a)$ is the $n$-th derivative of $\mathcal{L}(H, h_i)$ evaluated at point $a$ and $R_N(h_i)$ is $N$-th remainder. Then, approximating $\mathcal{L}(H, h_i = 0)$ with a first-order Taylor polynomial where $h_i$ equals zero:

$$\mathcal{L}(H, h_i = 0) = \mathcal{L}(H, h_i) - \frac{\partial\mathcal{L}(H, h_i)}{\partial h_i}h_i - R_1(h_i). \quad (4)$$

The remainder $R_1$ can be represented in the form of Lagrange:

$$R_1(h_i) = \frac{\partial^2\mathcal{L}(H, h_i)}{\partial^2\delta h_i}h_i^2, \quad (5)$$

where $\delta \in (0, 1)$. Considering the use of ReLU activation function (Glorot et al., 2011) in the model, the first derivative of loss function tends to be constant, so the second order term tends to be zero in the end of training. Thus, we can ignore the remainder and get the importance evaluation function as follows:

$$\Theta_{\text{TE}}(i) = |\Delta\mathcal{L}(h_i)| = \left|\frac{\partial\mathcal{L}(H, h_i)}{\partial h_i}h_i\right|. \quad (6)$$

In practice, we need to accumulate the product of the activation and the gradient of the objective function w.r.t to the activation, which is easily computed during back-propagation. Finally, the evaluation function is shown as:

$$\Theta^m_{\text{TE}}(i^l) = \frac{1}{T_m}\sum_t \left|\frac{\delta\mathcal{L}(H, h^l_i)}{\delta h^l_i}h^l_i\right|, \quad (7)$$

where $h^l_i$ is the activation value of the $i$-th neuron of $l$-th layer and $T_m$ is the number of the training examples of language pair $m$. The criterion is computed on the data of language pair $m$ and averaged over $T_m$.

**Absolute Value** We adopt the magnitude-based neuron importance evaluation scheme (See et al., 2016), where the absolute value of each neuron's activation value is treated as the importance:

$$\Theta^m_{\text{AV}}(i^l) = \frac{1}{T_m}\sum_t |h^l_i|. \quad (8)$$

The notations in the above equation are the same as those in the Equation 7. After the importance of each neuron is evaluated on the combined data, we need to determine the role of each neuron in the fine-tuning step following the method in the next section.

### 3.2 Neuron Allocation

In this step, we should determine which neurons are shared across all the language pairs and which neurons are shared only for some specific language pairs.

**General Neurons** According to the overall importance $\mathcal{I}(i)$ in Equation 1, the value correlates positively with how important the neuron is to all languages. Therefore, we rank the neurons in each layer based on the importance and make the top $\rho$ percentage as general neurons that are responsible for capturing general knowledge.

**Language-specific Neurons** Next, we regard other neurons except for the general neurons as the language-specific neurons and determine which language pair to assign them to. To achieve this, we compute an importance threshold for each neuron:

$$\lambda(i) = k \times \max(\Theta^m(i)), \quad m \in \{1, \ldots, M\}, k \in [0, 1] \tag{9}$$

, where $\max(\Theta^m(i))$ denotes the maximum importance of this neuron in all language pairs and $k$ is a hyper-parameter. The neuron will be assigned to the language-pairs whose importance is larger than the threshold. When the importance of neurons is determined, the number of language pairs associated with each neuron can be adjusted according to $k$. The smaller the $k$, the more language-pairs will be associated with the specific neurons. In this way, we flexibly determine the language pairs assigned to each neuron according to its importance in different languages. Note that the neuron allocation is based on the importance of *language pair*. We have also tried other allocation variants, e.g., based on the source language, target language, and find that the language pair-based method is the best among of these methods. The detailed results are listed in Appendix A.

After this step, the model is continually fine-tuned on the combined multilingual data. If the training data is from a specific language pair, only the general neurons and the language-specific neurons for this language pair will participate in the forward computation and the parameters associated with them will be updated during the backward propagation.

## 4 Experiments

### 4.1 Data Preparation

In this section, we describe the datasets using in our experiments on many-to-many and one-to-many multilingual translation scenarios.

**Many-to-Many** For this translation scenario, we test our approach on IWSLT-17[1] translation datasets, including English, Italian, Romanian, Dutch (briefly, En, It, Ro, Nl). We experimented in eight directions, including It↔En, Ro↔En, Nl↔En, and It↔Ro, with 231.6k, 220.5k, 237.2k, and 217.5k data for each language pair. We choose test2016 and test2017 as our development and test set, respectively. Sentences of all languages were tokenized by the Moses scripts[2] and further segmented into subword symbols using Byte-Pair Encoding (BPE) rules (Sennrich et al., 2016) with 40K merge operations for all languages jointly.

**One-to-Many** We evaluate the quality of our multilingual translation models using training data from the Europarl Corpus[3], Release V7. Our experiments focus on English to twelve primary languages: Czech, Finnish, Greek, Hungarian, Lithuanian, Latvian, Polish, Portuguese, Slovak, Slovene, Swedish, Spanish (briefly, Cs, Fi, El, Hu, Lt, Lv, Pl, Pt, Sk, Sl, Sv, Es). For each language pair, we randomly sampled 0.6M parallel sentences as training corpus (7.2M in all). The Europarl evaluation data set dev2006 is used as our validation set, while devtest2006 is our test set. For language pairs without available development and test set, we randomly split 1K unseen sentence pairs from the corresponding training set as the development and test data respectively. We tokenize and truecase the sentences with Moses scripts and apply a jointly-learned set of 90k BPE obtained from the merged source and target sides of the training data for all twelve language pairs.

### 4.2 Systems

To make the evaluation convincing, we re-implement and compare our method with four baseline systems, which can be divided into two categories with respect to the number of models. The multiple-model approach requires maintaining a dedicated NMT model for each language:

[1] https://sites.google.com/site/iwsltevaluation2017
[2] http://www.statmt.org/moses/
[3] http://www.statmt.org/europarl/

| | It→En | En→It | Ro→En | En→Ro | Nl→En | En→Nl | It→Ro | Ro→It | AVE | Para |
|---|---|---|---|---|---|---|---|---|---|---|
| **Individual** | 34.99 | 31.22 | 28.58 | 23.19 | 30.21 | 27.69 | 19.52 | 20.95 | 27.04 | 466.4M |
| **Multilingual** | 37.55 | 32.62 | 31.58 | 24.64 | 31.13 | 28.86 | 20.82 | 23.79 | 28.87 | **64.69M** |
| **+TS** | 38.11 | 33.46 | 31.82 | 24.96 | 32.04 | 30.06 | 21.43 | 23.59 | $29.43^{+0.56}$ | 121.42M |
| **+Adapter** | 38.25 | 34.16 | 32.07 | 25.08 | 32.56 | 29.66 | 21.18 | 24.26 | $29.65^{+0.78}$ | 77.43M |
| **Our Method-AV** | 38.07 | 34.15 | 32.17 | 26.00 | 32.21 | 30.11 | 21.96 | 24.46 | $29.89^{+1.02}$ | **64.69M** |
| **Our Method-TE** | **38.31** | **34.24** | **32.24** | **26.34** | **32.73** | **30.16** | **22.21** | **24.76** | $\mathbf{30.12}^{+1.25}$ | **64.69M** |

Table 1: BLEU scores on the many-to-many translation tasks. 'AVE' denotes the average BLEU of the eight test sets and 'Para' denotes the number of parameters of the whole model. 'Para' of the Individual system is the sum of the models for the eight language pairs with 58.3M parameters for each model.

| | Cs | El | Es | Fi | Hu | Lt | Lv | Pl | Pt | Sk | Sl | Sv | AVE | Para |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Individual** | 36.14 | 39.86 | 41.16 | 22.95 | 31.75 | 32.31 | 38.12 | 32.95 | 35.57 | 40.51 | 43.83 | 33.23 | 35.70 | 746.76M |
| **Multilingual** | 37.87 | 40.34 | 41.58 | 23.03 | 31.10 | 33.11 | 39.22 | 32.67 | 36.20 | 42.05 | 44.76 | 33.16 | 36.26 | **90.42M** |
| **+TS** | 37.70 | 40.70 | 42.05 | 23.28 | 31.78 | 32.90 | 39.48 | 33.66 | 36.09 | 42.03 | 44.29 | 33.14 | $36.43^{+0.17}$ | 273.77M |
| **+Adapter** | 38.11 | 40.23 | 41.83 | 23.66 | 32.00 | 33.49 | 39.87 | 32.85 | 36.25 | 42.00 | 44.63 | 32.90 | $36.49^{+0.23}$ | 109.54M |
| **Our Method-AV** | 37.84 | **40.75** | 42.16 | 23.71 | 31.40 | 33.56 | 39.95 | 33.23 | 36.56 | 42.09 | 45.27 | 33.38 | $36.66^{+0.40}$ | **90.42M** |
| **Our Method-TE** | **38.21** | 40.70 | 42.22 | **23.74** | 31.32 | 33.55 | 39.78 | 32.94 | **36.58** | 41.91 | 44.94 | 33.07 | $36.58^{+0.32}$ | **90.42M** |
| **+Expansion** | 38.03 | 40.59 | **42.28** | 23.73 | **32.47** | **34.12** | **40.12** | **33.95** | 36.41 | **42.44** | **45.30** | **33.43** | $\mathbf{36.91}^{+0.65}$ | 102.14M |

Table 2: BLEU scores on one-to-many translation tasks. 'Para' of the Individual system is 62.23M for each language pair. The denotations represent the same meaning as in Table 1.

**Individual** A NMT model is trained for each language pair. Therefore, there are $N$ different models for $N$ language pairs.

The unified model-based methods handle multiple languages within a single unified NMT model:

**Multilingual** (Johnson et al., 2017) Handling multiple languages in a single transformer model which contains one encoder and one decoder with a special language indicator *lang* added to the input sentence.

**+TS** (Blackwood et al., 2018) This method assigns language-specific attention modules to each language pair. We implement the target-specific attention mechanism because of its excellent performance in the original paper.

**+Adapter** (Bapna and Firat, 2019) This method injects tiny adapter layers for specific language pairs into the original MNMT model. We set the dimension of projection layer to 128 and train the model from scratch.

**Our Method-AV** Our model is trained just as the Approach section describes. In this system, we adopt the absolute value based method to evaluate the importance of neurons across languages.

**Our Method-TE** This system is implemented the same as the system *Our Method-AV* except that we adopt the Taylor Expansion based evaluation method as shown in Equation 7.

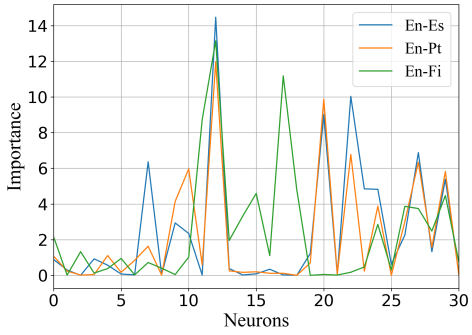
**+Expansion** To make a fair comparison, we set the size of Feed Forward Network to 3000 to expand the model capacity up to the level of other baselines, and then apply our Taylor Expansion based method to this model.

### 4.3 Details

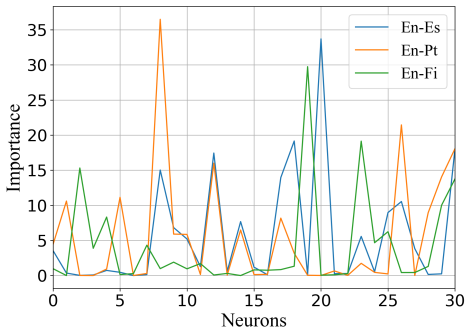
For fair comparisons, we implement the proposed method and other contrast methods on the advanced Transformer model using the open-source toolkit *Fairseq-py* (Ott et al., 2019). We follow Vaswani et al. (2017) to set the configurations of the NMT model, which consists of 6 stacked encoder/decoder layers with the layer size being 512. All the models were trained on 4 NVIDIA 2080Ti GPUs where each was allocated with a batch size of 4,096 tokens for one-to-many scenario and 2,048 tokens for the many-to-many scenario. We train the baseline model using Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. The proposed models are further trained with corresponding parameters initialized by the pre-trained baseline model. We vary the hyper-parameter $\rho$ that controls the proportion of general neurons in each module from $80\%$ to $95\%$ and set it to $90\%$ in our main experiments according to the performance. The detailed results about this hyper-parameter are list in Appendix B. We set the hyper-parameter $k$ to $0.7$ and do more analysis on it in Section 5.3. For evaluation, we use beam search with a beam size of 4 and length penalty $\alpha = 0.6$.

### 4.4 Results

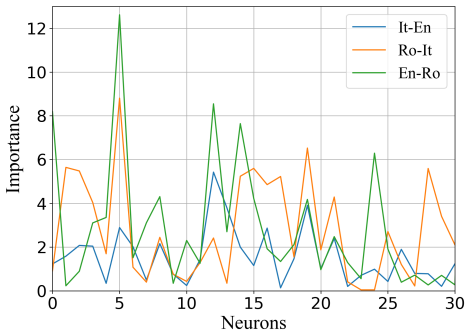
The final translation is detokenized and then the quality is evaluated using the 4-gram case-sensitive



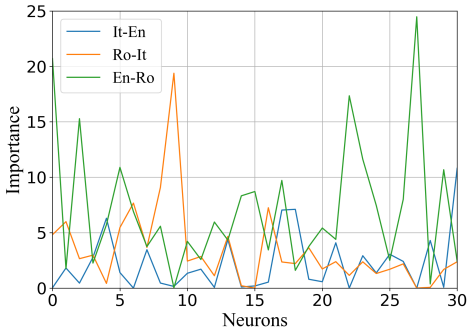

(a) O2M-Enc-6-FFN




(b) O2M-Dec-6-FFN




(c) M2M-Enc-6-FFN




(d) M2M-Dec-6-FFN

Figure 2: Importance distribution of neurons computed by Taylor Expansion in each module. For example, 'O2M-Enc-6-FFN' represents the importance of the feed forward network in the 6-th encoder layer.

BLEU (Papineni et al., 2002) with the *SacreBLEU* tool (Post, 2018).[4]

**Many-to-Many** The results are given in Table 1. We can see that the improvements brought by +TS and +Adapter methods are not large. For the +TS method, attention module may be not essential to capture language-specific knowledge, and thus it is difficult to converge to good optima. For the +Adapter method, adding an adapter module to the end of each layer may be not appropriate for some languages and hence has a loose capture to the specific features. In all language pairs, our method based on Taylor Expansion outperforms all the baselines in the datasets. Moreover, the parameters in our model are the same as the Multilingual system and less than other baselines.

**One-to-Many** The results are given in Table 2, our method exceeds the multilingual baseline in all language pairs and outperforms other baselines in most language pairs without capacity increment. When we expand the model capacity to the level of +Adapter, our approach can achieve better translation performance, which demonstrates the effectiveness of our method. Another finding is that the results of the individual baseline are worse than other baselines. The reason may be the training data is not big enough, individual baseline can not get a good enough optimization on 0.6M sentences, while the MNMT model can be well trained with a total of 7.2M data.

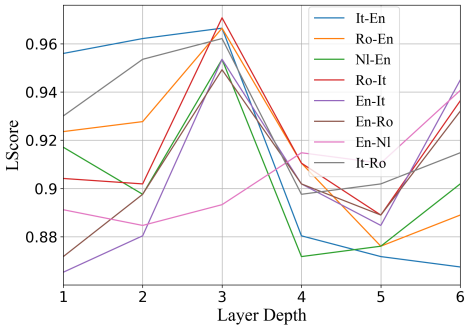
## 5 Analysis

### 5.1 Neuron Importance for Different languages

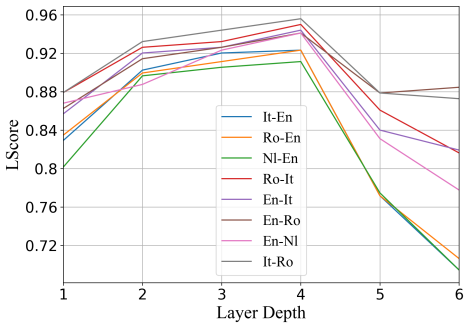
In our method, we allocate neurons based on their importance for different languages. The rationality behind this mechanism is that different neurons should have distinct importance values so that these neurons can find their relevant language pairs. Therefore, we show the importance of neurons computed by Taylor Expansion in different modules for the one-to-many (O2M) and many-to-many (M2M) translation tasks. For clarity and convenience, we only show the importance values of three language pairs in the sixth layer of encoder and decoder.

The results of O2M are shown in Figure 2(a) and Figure 2(b), and the language pairs are En→Es, En→Pt, and En→Fi. The first two target languages

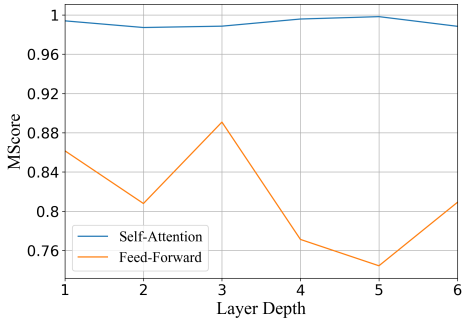
[4]BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a +version.1.4.14



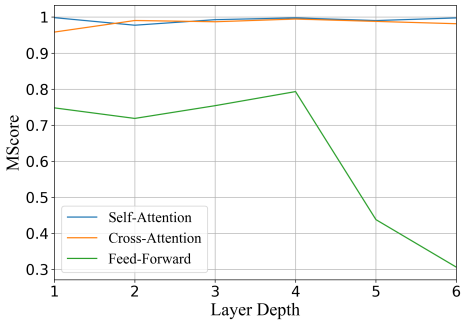

(a) Encoder




(b) Decoder




(c) Encoder




(d) Decoder

Figure 3: The distribution of the language-specific neurons in the encoder and decoder. The importance of neurons is computed by Taylor Expansion. The first two sub-figures show the proportion of specific neurons for different language pairs, while the last two sub-figures show the proportion of specific neurons in different modules.

are Spanish and Portuguese, both of which belong to the Western Romance, the Romance branch of the Indo-European family, while the last one is Finnish, a member of the Finnish-Ugric branch of the Ural family. As we can see, the importance of Spanish and Portuguese are always similar in most neurons, but there is no obvious correlation between Finnish and the other two languages. It indicates that similar languages are also similar in the distribution of the neuron importance, which implies that the common features in similar languages can be captured by the same neurons.

The results of M2M are shown in Figure 2(c) and Figure 2(d), and the language pairs are It→En, Ro→It, and En→Ro, whose BLEU scores are 0.67, 1, and 1.7 higher than the multilingual baseline, respectively. In most neurons, the highest importance value is twice as high as the lowest and this high variance of importance provides the theoretical basis for later neuron allocation. Moreover, we can see a lot of importance peaks of the two language pairs: Ro→It and En→Ro, which means that these neurons are especially important for generating the translation for these language pairs. However, the fluctuation of It→En is flat with almost no peaks, which means only a few neurons are specific to this language pair. This may be the reason why some language pairs have higher improvements, while some have lower improvements.

### 5.2 Distribution of the Language-specific Neurons

Except for the general neurons shared by all the language pairs, our method allocates other neurons to different language pairs based on their importance. These language-specific neurons are important for preserving the language-specific knowledge. To better understand the effectiveness of our method, we will show how these specific neurons are distributed in the model.

To evaluate the proportion of language-specific neurons for different language pairs at each layer, we introduce a new metric, LScore, formulated as:

$$\text{LScore}(l, m) = \frac{\tilde{I}_l^m}{\tilde{I}_l}, m \in \{1, \ldots, M\} \quad (10)$$

where $\tilde{I}_l^m$ denotes the number of neurons allocated to language pair $m$ in the $l$-th layer, and $\tilde{I}_l$ denotes the total number of the language-specific neurons in the $l$-th layer. The larger the LScore, the more neurons allocated to the language pair $m$. We also

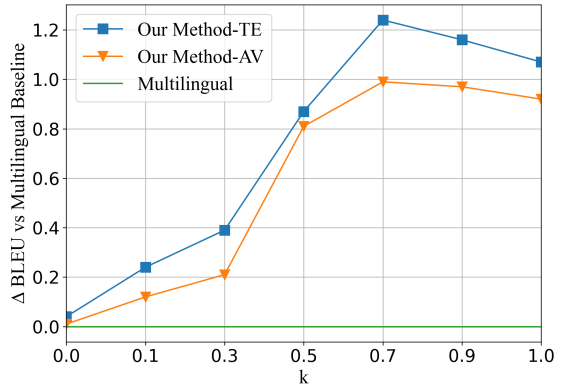



Figure 4: The average Δ BLEU over the Multilingual baseline with different hyper-parameters $k$ on the many-to-many translation task.

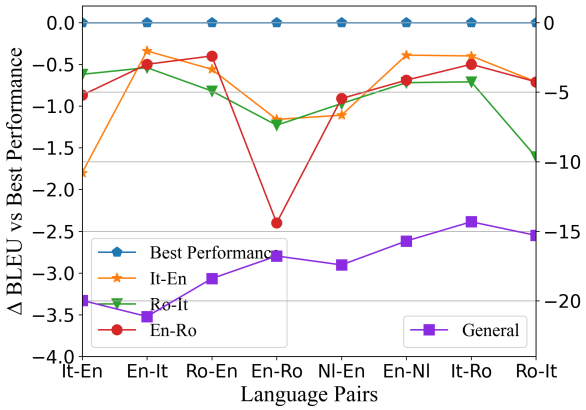



Figure 5: Δ BLEU over best performance when erasing the general or language-specific neurons randomly on the many-to-many translation task.

introduce a metric to evaluate the average proportion of language-specific neurons of each language in different modules, which formulated as:

$$\text{MScore}(l, f) = \frac{1}{M} \sum_{m=0}^{M} \frac{\tilde{I}_{l,f}^{m}}{\tilde{I}_{l,f}}, m \in \{1, \ldots, M\} \tag{11}$$

where $\tilde{I}_{l,f}^{m}$ denotes the number of specific neurons for language pair $m$ of in the $f$ module of the $l$-th layer and $M$ denotes the total number of the language pair. The larger the MScore is, the more specific neurons are allocated to different language pairs in this module.

As shown in Figure 3(a) and Figure 3(b), the language pairs have low LScores at the top and bottom layers and high LScores at the middle layers of both the encoder and decoder. The highest LScore appears at the third or fourth layers, which indicates that the neuron importance of different language pairs is similar and the neurons of the middle layers are shared by more languages. As a contrast, the bottom and top layers will be more specialized for different language pairs. Next, from Figure 3(c) and Figure 3(d), we can see the MScores of the attention modules are almost near 1.0, which means neurons in self attention and cross attention are almost shared across all language pairs. However, the MScores of Feed Forward Network (FFN) gradually decrease as layer depth increases and it shows that the higher layers in FFN are more essential for capturing the language-specific knowledge.

### 5.3 Effects of the Hyper-parameter $k$

When the importance of neurons for different languages is determined, the number of language pairs associated with each neuron can be adjusted according to $k$. When $k = 1.0$, the threshold is $\max(\Theta^{m}(i))$ as computed by Equation 9, so the neurons will only be allocated to the language pair with the highest importance, and when $k = 0$, the threshold is 0 so the neurons will be shared across all language pairs just like the Multilingual baseline. To better show the overall impact of the hyper-parameter $k$, we vary it from 0 to 1 and the results are shown in Figure 4. As we can see, the translation performance of the two proposed approaches increases with the increment of $k$ and reach the best performance when $k$ equals 0.7. As $k$ continues to increase, the performance deteriorates, which indicates that the over-specific neurons are bad at capturing the common features shared by similar languages and will lead to performance degradation.

### 5.4 The Specific and General knowledge

The main idea of our method is to let the general knowledge and the language-specific knowledge be captured by different neurons of our method. To verify whether this goal has been achieved, we conduct the following experiments. For the general knowledge, we randomly erase 20% general neurons of the best checkpoint of our method, which means we mask the output value of these neurons to 0, then generate translation using it. For language-specific knowledge, we randomly erase 50% specific neurons and then generate translation.

As shown in Figure 5, when the general neurons are erased, the BLEU points of all the language pairs drop a lot (about 15 to 20 BLEU), which indicates general neurons do capture the general knowledge across languages. For specific neurons,

we show three language pairs for the sake of convenience. We can see that when the neurons associated with the current language pair are erased, the performance of this language pair decreases greatly. However, the performance of other language pairs only declines slightly, because the specific knowledge captured by these specific neurons are not so important for other languages.

## 6 Related Work

Our work closely relates to language-specific modeling for MNMT and model pruning which we will recap both here. Early MNMT studies focus on improving the sharing capability of individual bilingual models to handle multiple languages, which includes sharing encoders (Dong et al., 2015), sharing decoders (Zoph et al., 2016), and sharing sublayers (Firat et al., 2016). Later, Ha et al. (2016) and Johnson et al. (2017) propose an universal MNMT model with a target language token to indicate the translation direction. While this paradigm fully explores the general knowledge between languages and hard to obtain the specific knowledge of each language (Tan et al., 2019; Aharoni et al., 2019), the subsequent researches resort to Language-specific modeling, trying to find a better trade-off between sharing and specific. Such approaches involve inserting conditional language-specific routing layer (Zhang et al., 2021), specific attention networks (Blackwood et al., 2018; Sachan and Neubig, 2018), adding task adapters (Bapna and Firat, 2019), and training model with different language clusters (Tan et al., 2019), and so on. However, these methods increase the capacity of the model which makes the model bloated.

Moreover, our method is also related to model pruning, which usually aims to reduce the model size or improve the inference efficiency. Model pruning has been widely investigated for both computer vision (CV) (Luo et al., 2017) and natural language processing (NLP) tasks. For example, See et al. (2016) examines three magnitude-based pruning schemes, Zhu and Gupta (2018) demonstrates that large-sparse models outperform comparably-sized small-dense models, and Wang et al. (2020a) improves the utilization efficiency of parameters by introducing a rejuvenation approach. Besides, Lan et al. (2020) presents two parameter reduction techniques to lower memory consumption and increase the training speed of BERT.

## 7 Conclusion

The current standard models of multilingual neural machine translation fail to capture the characteristics of specific languages, while the latest researches focus on the pursuit of specific knowledge while increasing the capacity of the model and requiring fine manual design. To solve the problem, we propose an importance-based neuron allocation method. We divide neurons to general neurons and language-specific neurons to retain general knowledge and capture language-specific knowledge without model capacity incremental and specialized design. The experiments prove that our method can get superior translation results with better general and language-specific knowledge.

## Acknowledgments

We thank all the anonymous reviewers for their insightful and valuable comments. This work was supported by National Key R&D Program of China (NO. 2017YFE0192900).

## References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3874–3884. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1538–1548. Association for Computational Linguistics.

Graeme W. Blackwood, Miguel Ballesteros, and Todd Ward. 2018. Multilingual neural machine translation with task-specific attention. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3112–3122. Association for Computational Linguistics.

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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1723–1732.

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2020. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. *CoRR*, abs/2004.06575.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 866–875.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1243–1252.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 315–323. JMLR.org.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O. K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 344–354.

Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *CoRR*, abs/1611.04798.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1700–1709.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. 2017. Thinet: A filter level pruning method for deep neural network compression. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 5068–5076. IEEE Computer Society.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Devendra Singh Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of*

*the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 261–271. Association for Computational Linguistics.

Abigail See, Minh-Thang Luong, and Christopher D. Manning. 2016. Compression of neural machine translation models via pruning. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 291–301. ACL.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 963–973.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Raúl Vázquez, Alessandro Raganato, Jörg Tiedemann, and Mathias Creutz. 2019. Multilingual NMT with a language-independent attention bridge. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 33–39, Florence, Italy. Association for Computational Linguistics.

Yong Wang, Longyue Wang, Victor O. K. Li, and Zhaopeng Tu. 2020a. On the sparsity of neural machine translation models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1060–1066. Association for Computational Linguistics.

Yong Wang, Longyue Wang, Shuming Shi, Victor O. K. Li, and Zhaopeng Tu. 2020b. Go from the general to the particular: Multi-domain translation with domain transformation networks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9233–9241. AAAI Press.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

Michael Zhu and Suyog Gupta. 2018. To prune, or not to prune: Exploring the efficacy of pruning for model compression. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1568–1575. The Association for Computational Linguistics.

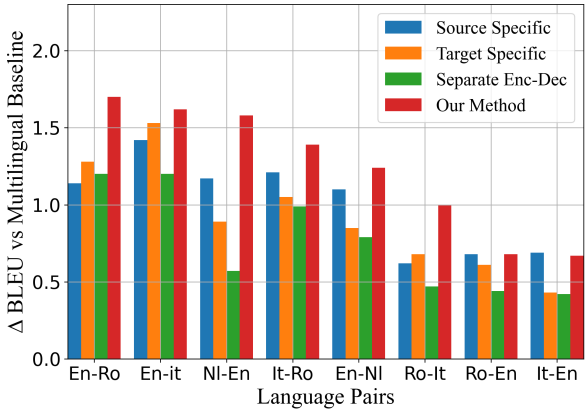



Figure 6: Δ BLEU over Multilingual baseline on many-to-many translation.

## A Performance on Different Varieties

In the proposed method we allocate neurons based on importance of language pair. There are three varieties of our method: (a) Source-Specific, share all neurons according to the source language only; (b) Target-Specific, share all neurons according to the target language only; (c) Separate Enc-Dec, Encoder neurons are shared according to the source language and decoder neurons are shared according to the target language. Note that (c) is different from our method since (c) is separate neurons to two parts (encoder and decoder) and then connect specific neurons of the two parts to form a whole, while our method is directly based on language pairs.

As shown in Figure 6, we compare our Taylor Expansion method with the other three varieties. Our approach outperforms other varieties on almost all language pairs, and the performance of the language-pair based approach is undoubtedly the best. The second is based on the target language and the source language. Worst of all are the separated encoder-decoder, which may be due to the mismatch between the neurons of the encoder and decoder when they are reconnected.

## B Effects of the Hyper-parameter $\rho$

We conducted several experiments on $\rho$ to determine the optimal hyper-parameter, so as to determine the proportion of universal neurons. As shown in Table 3, when $\rho = 90\%$ the model gets the best translation result and reach best trade-off between general and language-specific neurons.

| | It→En | En→It | Ro→En | En→Ro | Nl→En | En→Nl | It→Ro | Ro→It | AVE |
|---|---|---|---|---|---|---|---|---|---|
| $\rho = 80\%$ | 38.3 | 34.05 | 32.11 | 26.01 | 32.24 | 30.12 | 21.96 | 24.39 | 29.94 |
| $\rho = 90\%$ | **38.31** | **34.15** | **32.24** | **26.34** | **32.73** | **30.16** | **22.21** | **24.76** | **30.11** |
| $\rho = 95\%$ | 38.28 | 33.82 | 32.05 | 25.74 | 31.97 | 29.51 | 21.56 | 24.19 | 29.64 |

Table 3: BLEU scores on many-to-many translation tasks when $k = 0.7$