ME-MD: An Effective Framework for Neural Machine Translation with Multiple Encoders and Decoders

Jinchao Zhang[1] **Qun Liu**[3,1] **Jie Zhou**[2]
[1]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS.
[2]Baidu Research - Institute of Deep Learning, Baidu Inc.,Beijing,China
[3]ADAPT Centre, School of Computing, Dublin City University
{zhangjinchao,liuqun}@ict.ac.cn, zhoujie01@baidu.com

Abstract

The encoder-decoder neural framework is widely employed for Neural Machine Translation (NMT) with a single encoder to represent the source sentence and a single decoder to generate target words. The translation performance heavily relies on the representation ability of the encoder and the generation ability of the decoder. To further enhance NMT, we propose to extend the original encoder-decoder framework to a novel one, which has multiple encoders and decoders (**ME-MD**). Through this way, multiple encoders extract more diverse features to represent the source sequence and multiple decoders capture more complicated translation knowledge. Our proposed ME-MD framework is convenient to integrate heterogeneous encoders and decoders with multiple depths and multiple types. Experiment on Chinese-English translation task shows that our ME-MD system surpasses the state-of-the-art NMT system by **2.1** BLEU points and surpasses the phrase-based Moses by **7.38** BLEU points. Our framework is general and can be applied to other sequence to sequence tasks.

1 Introduction

The encoder-decoder framework [Kalchbrenner and Blunsom, 2013; Cho *et al.*, 2014; Sutskever *et al.*, 2014] is widely exploited for Neural Machine Translation. In this framework, an encoder compresses the source sentence to a distribution representation and a decoder generates target words one by one regarding the source representation. Compared with the Statistical Machine Translation (SMT), NMT models the translation knowledge through training a single network in the end-to-end style and gets ride of constructing several sub-components separately.

Plenty of approaches are proposed to enhance the NMT performance, such as attention mechanisms [Bahdanau *et al.*, 2015; Luong *et al.*, 2015a; Meng *et al.*, 2016], effective connections [Zhou *et al.*, 2016; Wu *et al.*, 2016], coverage models [Tu *et al.*, 2016], addressing rare words [Jean *et al.*, 2015; Luong *et al.*, 2015b; Sennrich *et al.*, 2016; Chung *et al.*, 2016], joint training [Dong *et al.*, 2015; Luong *et al.*, 2016; Firat *et al.*, 2016; Zoph and Knight, 2016], external memory [Wang *et al.*, 2016] and sentence-level training [Shen *et al.*, 2016].

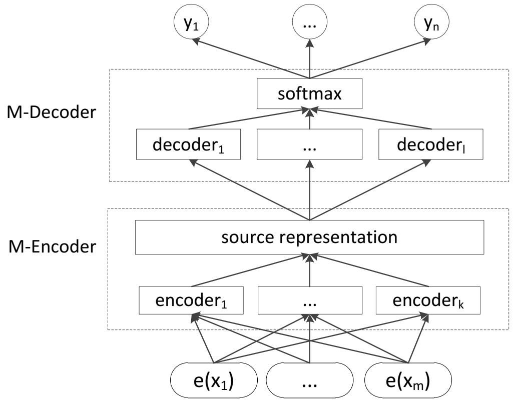
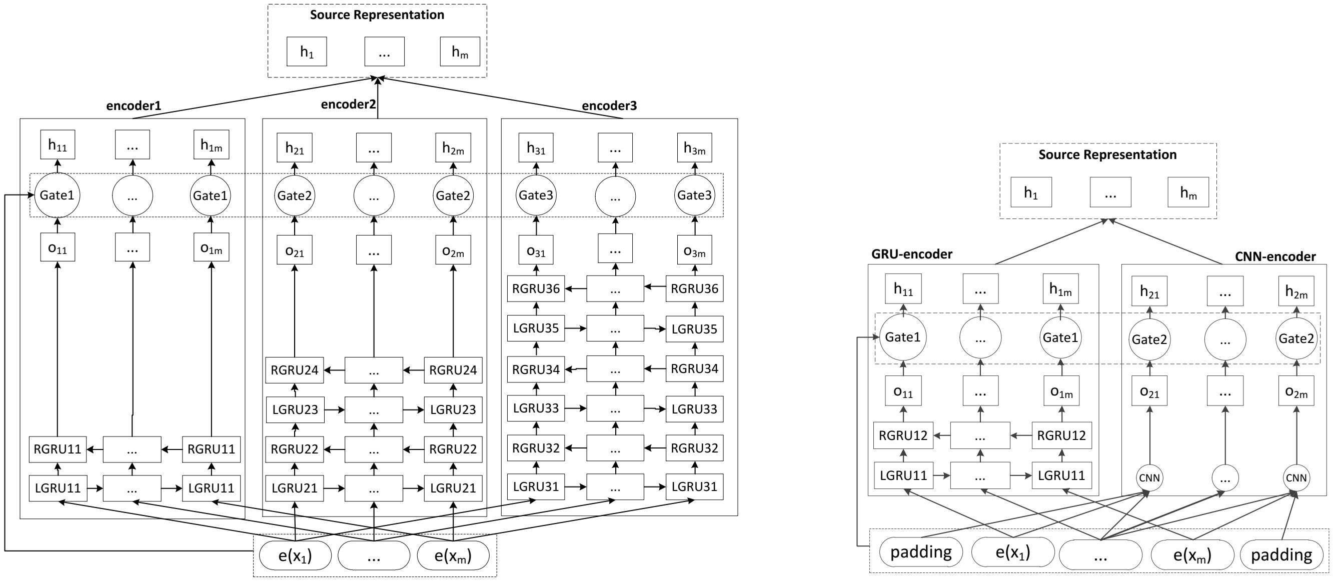


Figure 1: The general architecture of the proposed ME-MD framework. The architecture consists of two modules: M-Encoder and M-Decoder. Compared with the encoder-decoder framework, ME-MD exploits multiple encoders and decoders.

The translation performance heavily relies on the source sentence representation ability of the encoder and the target sentence generation ability of the decoder. To further enhance NMT, we propose a novel framework named as "**ME-MD** (multiple encoders and multiple decoders)", which exploits multiple encoders to represent the source sequence and multiple decoders to generate target words. These encoders and decoders are allowed to possess different depths or multiple types. The basic idea is that multiple encoders provide more comprehensive source representation and multiple decoders capture more complicated translation knowledge. We implement several ME-MD systems and carry experiments on the Chinese-English translation task. Experimental results show that ME-MD systems outperform the encoder-decoder baseline by large margins. Our best system surpasses the state-of-the-art NMT system by 2.1 BLEU points and exceeds phrase-based Moses by 7.38 BLEU points. We also validate that our approach benefits more from the architecture change rather than making the network wider and deeper. Although, we conduct the experiment on machine translation task, our framework is general and can be applied to other sequence to



(a)

(b)

Figure 2: (a) is an multi-depth M-Encoder with three sub-encoders. Each encoder has unique depth. The source sentence is compressed by encoders respectively and the distributed representations are combined into one as the comprehensive representation of the source sentence. (b) is an multi-type M-Encoder with one GRU-based encoder and one CNN-based encoder. The representations from GRU-based encoder and CNN-based encoder are combined into one to represent the source sentence.

sequence tasks.

2 Neural Machine Translation

We briefly introduce the NMT architecture [Bahdanau *et al.*, 2015] that our systems build on. Formally, given a source sentence $\mathbf{x} = \mathbf{x}_1, ..., \mathbf{x}_m$ and a target sentence $\mathbf{y} = \mathbf{y}_1, ..., \mathbf{y}_n$, NMT models the translation probability as

$$P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{n} P(\mathbf{y}_t|\mathbf{y}_{<t}, \mathbf{x}), \tag{1}$$

where $\mathbf{y}_{<t} = \mathbf{y}_1, ..., \mathbf{y}_{t-1}$.

The NMT system primarily consists of two parts: the encoder and the decoder. For convenient explanation, we attribute the attention mechanism as a sub-component of the decoder. The encoder compresses the source sentence into the distribution representation and the decoder generates target words one by one regarding the source representation, as

$$\begin{aligned} \mathbf{h} = \{\mathbf{h}_1, ..., \mathbf{h}_m\} &= Encoder(\mathbf{x}) \\ \mathbf{y} = \{\mathbf{y}_1, ..., \mathbf{y}_n\} &= Decoder(\mathbf{h}), \end{aligned} \tag{2}$$

where $\mathbf{h}$ is the source representation. The generation probability of $\mathbf{y}_t$ is computed as

$$\begin{aligned} q &= g(\mathbf{y}_{t-1}, \mathbf{c}_t, \mathbf{s}_t) \\ P(\mathbf{y}_t|\mathbf{y}_{<t}, \mathbf{x}) &= softmax(q), \end{aligned} \tag{3}$$

where q is the context to predict the target word, $g(\cdot)$ is a linear function and $\mathbf{s}_t$ is the hidden state of decoder which represents the translation status. The attention $\mathbf{c}_t$ denotes the related source words for generating $\mathbf{y}_t$ and is computed as the weighted-sum of source representation $\mathbf{h}$ upon an alignment vector α_t shown in Eq.(4) where the $align(\cdot)$ function is a feed-forward network with the $softmax$ normalization.

$$\begin{aligned} \mathbf{c}_t &= attention(\mathbf{s}_{t-1}, \mathbf{h}) \\ &= \sum_{i=1}^{m} \alpha_{t,i} \mathbf{h}_i \\ \alpha_{t,i} &= align(\mathbf{s}_{t-1}, \mathbf{h}_i) \\ &= softmax(\mathbf{v}_\alpha^T tanh(\mathbf{W}_\alpha \mathbf{s}_{t-1} + \mathbf{U}_\alpha \mathbf{h}_j)). \end{aligned} \tag{4}$$

The hidden state $\mathbf{s}_t$ is updated as

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{y}_{t-1}, \mathbf{c}_t), \tag{5}$$

where $f(\cdot)$ is a gated hidden unit.

In recent, a varietal attention mechanism[1] is implemented as

$$\begin{aligned} \widetilde{\mathbf{s}}_t &= f_1(\mathbf{s}_{t-1}, \mathbf{y}_{t-1}), \\ \alpha_{t,j} &= align(\widetilde{\mathbf{s}}_t, \mathbf{h}_j), \\ \mathbf{s}_t &= f_2(\widetilde{\mathbf{s}}_t, \mathbf{c}_t), \end{aligned} \tag{6}$$

where $f_1(\cdot)$ and $f_2(\cdot)$ are recurrent functions. We adopt this varietal attention mechanism in our NMT systems.

3 ME-MD Framework

We aim to enhance NMT through incorporating multiple encoders and decoders. Our intuition is that multiple encoders provide comprehensive source representations and multiple decoders capture complicated translation knowledge.

[1] https://github.com/nyu-dl/dl4mt-tutorial/tree/master/session2

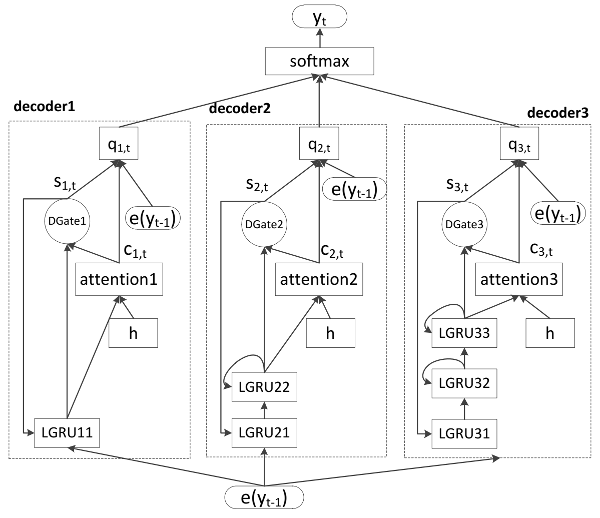


Figure 3: The architecture of a multi-depth M-Decoder with three decoders. The depth of decoders are 1, 2 and 3, respectively. Each decoder has independent RNN parameters and attention parameters. The outputs of three decoders are combined and feed into the *softmax* function to predict the target word.

3.1 General Architecture

The proposed ME-MD framework consists of two parts: **M-Encoder** and **M-Decoder**, as shown in Figure 1. The M-Encoder compresses the source sentence into the distribution representation as the source representation and the M-Decoder generates target sentence word by word upon the source representation.

Compared with the single encoder in the encoder-decoder framework, M-Encoder allows multiple encoders to represent the source sentence, respectively. All the source representation are combined to construct the ultimate source representation. Through incorporating diverse encoders, we expect to obtain a more comprehensive representation of the source sentence. In the M-Decoder, more than one decoders are incorporated to capture more complicated translation knowledge. The outputs of the decoders are combined before the *softmax* layer for predicting target words. The proposed ME-MD framework is flexible to integrate variable encoders and decoders and can be applied to other sequence to sequence tasks.

3.2 M-Encoder

Encoders in M-Encoder module can posses multi-depths and multi-types. The multi-depth M-Encoder combines a couple of encoders with different depths and the multi-type M-Encoder exploits encoders with different types.

We consider that multi-depth encoders can provide multiple level abstraction of the source sentence. Figure 2 (a) shows a multi-depth M-Encoder with three encoders, which depths are 2, 4 and 6, respectively. Without loss of generality, we take the "encoder 2" for detailed illustration. We exploit the left-to-right gated recurrent unit (LGRU) [Cho *et al.*, 2014] to forwardly compress the source sequence and the right-to-left gated recurrent unit (RGRU) to reversely compress the source sequence. Layers with different directions are alternately stacked with direct connections. After the input sequence is compressed by stacked GRU layers to the vector $\mathbf{o}_2 = \{\mathbf{o}_{21}, ..., \mathbf{o}_{2m}\}$, a gated unit is employed to combine original word embedding $e(\mathbf{x}_i)$ and $\mathbf{o}_{2i}$ as

$$\begin{aligned} z_{2i} &= sigmoid(W_{xz} \cdot e(\mathbf{x}_i) + W_{oz} \cdot \mathbf{o}_{2i} + b_z) \\ \tilde{h}_{2i} &= tanh(W_{xh} \cdot e(\mathbf{x}_i) + W_{oh} \cdot \mathbf{o}_{2i} + b_h) \\ \mathbf{h}_{2i} &= (1 - z_{2i}) * e(\mathbf{x}_t) + z_{2i} * \tilde{h}_{2i}, \end{aligned} \quad (7)$$

where W_{xz}, W_{oz}, W_{xh} and $W_{oh} \in \mathbb{R}^{d \times d}$ are the weight matrix parameters , b_z and $b_h \in \mathbb{R}^d$ are the bias parameters. For conveniently builds the network, we set the word dimension and hidden unit numbers to the identical value d. Three encoders produce three source representations as

$$\{\mathbf{h}_{11}, ..., \mathbf{h}_{1m}\}, \{\mathbf{h}_{21}, ..., \mathbf{h}_{2m}\}, \{\mathbf{h}_{31}, ..., \mathbf{h}_{3m}\}. \quad (8)$$

We combine the three representations with a feed-forward network as

$$\mathbf{h}_i = tanh(W_{h1} \cdot \mathbf{h}_{1i} + W_{h2} \cdot \mathbf{h}_{2i} + W_{h3} \cdot \mathbf{h}_{3i} + b), \quad (9)$$

in which W_{h1}, W_{h2}, and $W_{h3} \in \mathbb{R}^{d \times d}$ are the weight matrix parameters, $b_z \in \mathbb{R}^d$ is the bias parameter.

Figure 2 (b) shows a multi-type M-Encoder with two types of encoders to compress the source sentence. One encoder is the GRU-based network and another is the CNN-based network. The CNN encoder only possesses a convolutional layer with a fixed window size. The GRU encoder captures the global source representation and the CNN encoder focus on the local representation. The output of the CNN-encoder is computed as

$$o_{2i} = tanh(W_f \cdot [x_{i-\lfloor \frac{p}{2} \rfloor} : ... : x_{i+\lfloor \frac{p}{2} \rfloor}] + b), \quad (10)$$

where $W_f \in \mathbb{R}^{d \times p \times d}$ is the weight matrix parameters, b is the bias parameter and p is the convolutional window size. The gate computation and the ultimate source representation computation are identical to the multi-depth M-Encoder.

3.3 M-Decoder

The M-Decoder aims to enhance the generation ability of the decoder through integrating multiple decoders. Similar to the M-Encoder, the M-Decoder can also have multi-depth and multi-type. The multi-depth M-Decoder consists of a couple of decoders with different depths. The multi-type M-Decoder allows to exploit the variable attention mechanisms [Bahdanau *et al.*, 2015; Luong *et al.*, 2015a; Wu *et al.*, 2016] and multiple recurrent networks.

Figure 3 presents a multi-depth M-Decoder that contains three decoders with different depths. We take the "decoder2" for detailed description without loss of generality. We adopt the varietal decoder implementation in our NMT systems. Formally, the output $\mathbf{q}_{2,t}$ of the "decoder2" at time t is computed as following:

$$\begin{aligned} \widetilde{\mathbf{s}}_{21,t} &= LGRU21(\mathbf{s}_{2,t-1}, \mathbf{y}_{t-1}), \\ \widetilde{\mathbf{s}}_{22,t} &= LGRU22(\widetilde{\mathbf{s}}_{22,t-1}, \widetilde{\mathbf{s}}_{21,t}), \\ \mathbf{c}_{2,t} &= attention2(\widetilde{\mathbf{s}}_{22,t}, \mathbf{h}), \\ \mathbf{s}_{2,t} &= DGate2(\widetilde{\mathbf{s}}_{22,t}, \mathbf{c}_{2,t}), \\ \mathbf{q}_{2,t} &= g(\mathbf{s}_{2,t}, \mathbf{c}_{2,t}, \mathbf{y}_{t-1}), \end{aligned} \quad (11)$$

where $\widetilde{\mathbf{s}}_{21,t}$ and $\widetilde{\mathbf{s}}_{22,t}$ are outputs of the GRU layers, $\mathbf{c}_{2,t}$ is the related source context for generating target word y_t, function $attention(\cdot)$ is computed as in Eq.(4) and function $g(\cdot)$ is a linear one. The gate computation $DGate2(\widetilde{\mathbf{s}}_{22,t}, \mathbf{c}_{2,t})$ is

$$\begin{aligned}
&z = sigmoid(W_{cz} \cdot \mathbf{c}_{2,t} + W_{sz} \cdot \widetilde{\mathbf{s}}_{22,t} + b_{gz}),\\
&r = sigmoid(W_{cr} \cdot \mathbf{c}_{2,t} + W_{sr} \cdot \widetilde{\mathbf{s}}_{22,t} + b_{gr}),\\
&\widetilde{\mathbf{s}}_{2,t} = tanh((W_{ss} \cdot \widetilde{\mathbf{s}}_{22,t} + b_{ss}) * r + W_{cs} \cdot \mathbf{c}_{2,t}),\\
&\mathbf{s}_{2,t} = \widetilde{\mathbf{s}}_{22,t} * z + \widetilde{\mathbf{s}}_{2,t} * (1 - z),
\end{aligned} \tag{12}$$

in which z is the update gate, r is the reset gate, W_{cz}, W_{sz}, W_{cr}, W_{sr}, W_{ss} and W_{cz} are weight matrix parameters and b_{gz}, b_{gr} and b_{ss} are bias parameters. The outputs of three decoders are combined by a feed-forward network and feeded into $softmax$ function to predict the target word as

$$\begin{aligned}
&\mathbf{q}_t = tanh(W_{q1} \cdot \mathbf{q}_{1,t} + W_{q2} \cdot \mathbf{q}_{2,t} + W_{q3} \cdot \mathbf{q}_{3,t} + b_q),\\
&P(\mathbf{y}_t) = softmax(\mathbf{q}_t),
\end{aligned} \tag{13}$$

where W_{q1}, W_{q2} and W_{q3} are weight matrix parameters and b_q is the bias parameter.

Although we present several categories of M-Encoder and M-Decoder in this section, a large variety of encoders and decoders can be incorporated into our framework for its flexibility.

4 Experiment

We validate the effectiveness of the proposed framework on the Chinese-English translation task.

4.1 Data and Metrics

Our Chinese-English training corpus consists of 1.25M sentence pairs extracted from LDC corpora[2] with 27.9M Chinese words and 34.5M English words respectively. The 30K vocabularies cover approximately 97.7% and 99.3% words in Chinese and English respectively. We choose NIST 2002 dataset as the validation set. NIST 2003-2006 are used as test sets. The translation quality evaluation metric is the case-insensitive 4-gram BLEU[3] [Papineni *et al.*, 2002].

4.2 Systems

We implements 4 ME-MD systems and compare them with two baseline systems. The systems are listed as following:

1. **Moses** [Koehn *et al.*, 2007] is an open source phrase-based SMT baseline system with default settings. Words are aligned with GIZA++ [Och and Ney, 2003]. The 4-gram language model with modified Kneser-Ney smoothing is trained on the target portion of training data by SRILM [Stolcke and others, 2002].
2. **RNNsearch*** is our in-house implementation of RNNsearch [Bahdanau *et al.*, 2015] baseline system with the varietal attention mechanism. Different from the original model , we stack a forward GRU layer and a backward GRU layer with direct connection as a two layers encoder. The system can be regarded as the "1Encoders-1Decoder" ME-MD system and is the basis of other ME-MD systems.
3. **2Encoders-1Decoder** gets two GRU-based encoders and one GRU-based decoder. The depths of encoders are 2 and 4, respectively. The depth of the decoder is 1.
4. **3Encoders-1Decoder** has three GRU-based encoders and one GRU-based decoder. The depths of encoders are 2, 4 and 6, respectively. The depth of the decoder is 1.
5. **3Encoders-3Decoders** consists of three GRU-based encoders and three GRU-based decoders. The depths of encoders and decoders are 2, 4 and 6, respectively.
6. **GCEncoders-1Decoders** contains one GRU-based encoder and one CNN-based encoder. The depth of the GRU-based encoder is 2 and the convolutional window size of the CNN-based encoder is 3. The depth of the decoder is 1.

4.3 NMT Training

The sentence length for training NMT models is up to 50, while SMT model exploits whole training data without any restrictions. The word embedding dimension and the hidden unit numbers are set to 512. Square matrices are initialized in a random orthogonal way. Non-square matrices are initialized by sampling each element from the Gaussian distribution with mean 0 and variance 0.01^2. All bias are initialized to 0. Parameters are updated by Mini-batch Gradient Descent and learning rate is controlled by AdaDelta [Zeiler, 2012] with decay constant $\rho = 0.95$ and denominator constant $\epsilon = 1e - 6$. The batch size is 80. Dropout strategy [Srivastava *et al.*, 2014] is applied to the output layer with the dropout rate=0.5 to avoid over-fitting. The gradients of the cost function which have $L2$ norm larger than a predefined threshold 1.0 is normalized to the threshold to avoid gradients explosion [Pascanu *et al.*, 2013]. We exploit length normalization on candidate translations and the beam size for decoding is 12. The systems are implemented on the Theano library and trained with Tesla K40 GPUs.

4.4 Experiment Result

Table 1 shows the performance of each system. The 2Encoder-1Decoder system and the 3Encoder-1Decoder system surpass the RNNsearch* baseline by **0.52** and **1.90** BLEU points, from which we conclude that incorporating additional encoders can effectively improve the NMT performance. Through extending the number of decoders to three, we obtain further more **0.43** BLEU points that proves the effectiveness of the M-Decoder module. The GCEncoders-1Decoders system outperforms the RNNsearch* baseline by **1.19** BLEU points shows the CNN-based encoder improves the source representation ability of the M-Encoder. The GCEncoders-1Decoders system surpasses 2Encoders-1Decoder system by **0.67** BLEU points shows that CNN-based encoder provides

[2] The corpora includes LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

[3] `ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b.pl`

Id	Systems	MT03	MT04	MT05	MT06	Average
1	Moses	31.61	33.48	30.75	30.85	31.67
2	RNNsearch*	37.35	39.32	35.82	34.40	36.72$^{+(-,\ \mathbf{5.05})}$
3	2Encoders-1Decoder	37.88	39.77	36.28	35.02	37.24$^{+(\mathbf{0.52},\ \mathbf{5.57})}$
4	3Encoders-1Decoder	38.99	40.89	37.46	37.13	38.62$^{+(\mathbf{1.90},\ \mathbf{6.95})}$
5	3Encoders-3Decoders	38.93	41.69	38.24	37.34	39.05$^{+(\mathbf{2.33},\ \mathbf{7.38})}$
6	GCEncoders-1Decoders	38.84	40.62	36.46	35.73	37.91$^{+(\mathbf{1.19},\ \mathbf{6.24})}$

Table 1: BLEU-4 scores (%) on NIST test set 03-06 of Moses (default settings), RNNsearch*(1Encoder-1Decoder) and ME-MD systems (Id=3:6) with different numbers of encoders and decoders. The values in brackets are increases on RNNsearch* and Moses respectively. The result shows that ME-MD systems achieve significant improvements upon Moses and RNNsearch* baseline.

System	Length	MT03	MT04	MT05	MT06	Average
Coverage [Tu *et al.*, 2016]	80	-	-	32.73	32.47	-
MEMDEC [Wang *et al.*, 2016]	50	**36.16**	**39.81**	**35.91**	**35.98**	**36.95**
NMT$_{IA}$ [Meng *et al.*, 2016]	80	35.69	39.24	35.74	35.10	36.44
3Encoders-3Decoders	50	**38.93**	**41.69**	**38.24**	**37.34**	**39.05**$^{+2.1}$

Table 2: Comparisons with previous works on identical training corpora. Coverage is a basic RNNsearch model with a coverage model to alleviate the over-translation and under-translation problems. MEMDEC is to improve translation quality with the external memory. NMT$_{IA}$ exploits a readable and writable attention mechanism to keep track of interactive history in decoding. The vocabulary sizes of all work are 30K and maximum lengths of sentence differ. Our "3Encoders-3Decoders" system surpasses previous works by large margins and achieves the stat-of-the-art performance.

the diverse source representation.

We present the performance of previous works that employ identical training corpora in Table 2. Although we restrict the maximum length of sentence to 50, our model achieves the state-of-the-art performance on all test sets. Our ME-MD system outperforms previous work by at least **2.1** BLEU points.

4.5 Comparison with Deeper and Wider Networks

We carry more experiments to investigate whether our approach achieves improvements through just making the neural network seems deeper and wider or not. Table 3 shows the performance comparison between wider, deeper networks and the ME-MD systems and Figure 4 presents the training speeds of each system.

- **Wider Networks**. We enlarge the word embedding dimension and the hidden unit numbers to make networks wider. We achieve improvements as **0.79** BLEU points by extending the width from 512 to 1024 and obtain further more **0.32** BLEU points through set the width to 2048. However, the approach leads the rapid increase of the parameters and the dramatic decrease on the training speed. Compared with the wider networks, our approach provides larger improvements with less parameters and saves significant computation overhead.
- **Deeper Networks**. With the increasing of the depth, RNNsearch* achieves slight improvements or even obtains inferior performance. The reason is that it is difficult to train a very deep networks for the gradient propagation problem. Although our encoders and decoders are also very deep, we still achieve significant improvements for the shallow encoders is able to alleviate the gradient propagation problem. From the speed experiment, we observer that the speed of a ME-MD system is mainly depend on the deepest encoder and is close to the RNNsearch* that possess the same depth.

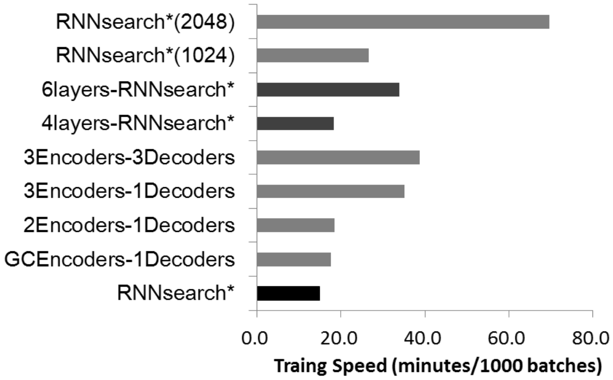


Figure 4: The training speed of each system. Enlarging the width of the RNNsearch* causes rapid decrease on training speed. The speed of a ME-MD system depends on its deepest encoder and is close to the RNNsearch* that has same depth. Compared with the deeper and wider networks, the ME-MD systems achieves significant improvements with modest increase on the training overhead.

The experimental results show that the improvements of our approach benefit more from the effective architecture than just introducing more parameters and ME-MD systems achieves significant improvements with modest increase on the training overhead.

5 Related Work

Our proposed ME-MD framework possesses multiple encoders and decoders, which is similar to the multi-task learning architectures that based on the encoder-decoder

Id	Systems	Dimensions	#Parameters	MT03	MT04	MT05	MT06	Average
1	RNNsearch*	512	53.7M	37.35	39.32	35.82	34.4	36.72
2	RNNsearch*	1024	122.6M	37.91	40.21	36.77	35.15	$37.51^{+0.79}$
3	RNNsearch*	2048	306.0M	37.82	40.65	36.36	36.49	$37.83^{+1.11}$
4	4layers-RNNsearch*	512	58.4M	37.31	39.02	36.08	34.62	$36.76^{+0.04}$
5	6layers-RNNsearch*	512	63.2M	36.08	38.51	35.08	33.61	$35.82^{-0.9}$
6	2Encoder-1Decoder	512	62.6M	37.88	39.77	36.28	35.02	$37.24^{+0.52}$
7	3Encoder-1Decoder	512	73.4M	38.99	40.89	37.46	37.13	$38.62^{+1.90}$
8	3Encoder-3Decoder	512	87.8M	**38.93**	**41.69**	**38.24**	**37.34**	**39.05**$^{+2.33}$

Table 3: From system 1 to system 3, we enlarge the word embedding dimension and hidden units number to construct wider network. Although, the approach provides improvements, the parameters scale increase rapidly, which leads to serious computational overhead. Our proposed ME-MD method offers greater improvements with less parameters growth. Comparing system 1, 4 and 5, the deeper networks slightly improve the translation quality or even produce inferior performance.

framework. Dong et al. [2015] proposed a unified network with one encoder and multiple decoders to simultaneously train couples of translation systems. These translation systems share the source sentence representation and generate target translations in different languages. Luong et al. [2016] presented a framework with multiple encoders and decoder for multiple-task sequence to sequence learning. The encoders and decoders are designed for multiple specific tasks, such as translation, parsing and image caption. Firat et al. [2016] proposed to share the attention mechanism for jointly training multi-lingual translation systems, in which encoders and decoders are employed for certain languages. The above mentioned works just activate one encoder and one decoder when deal with a certain task or translation direction. In our framework, all the encoders and decoders are utilized simultaneously, through which the translation quality is improved. The multi-source translation model with multiple encoders and attention mechanisms was proposed by Zoph and Knight [2016]. One encoder is applied to compress one kinds of source languages and all encoders outputs are combined to generate target translation. Compared with our work, their approach requires multi-way parallel corpus that is difficult to obtain.

6 Conclusion

We proposed an effective framework named "**ME-MD**" to enhance NMT performance with multiple encoders and decoders. Compared with the encoder-decoder framework, our approach enables to exploit multiple encoders and decoders with variable depths and types. The basic idea is that multiple encoders provide the more comprehensive representation of the source sentence and multiple decoders captures more complicated translation knowledge. To validate the effectiveness of our approach, we carry the experiment on Chinese-English translation task. We trained various networks architecture with M-Encoder and M-Decoder modules. Experiments show that ME-MD systems achieve significant improvements on translation quality over the basic encoder-decoder system and the phrase-based system by large margins. Through increasing the number and category of encoders and decoders, we acquire continuous improvements. The improvements benefit from the structural change of the original architecture. The comparison with previous works on identical training corpora shows our best model achieves the state-of-the-art performance. We also implemented the wider networks and found that enlarging the word embedding and hidden size can bring further improvements on translation quality. While, wider networks require enormous computing overhead, which needs longer training time and larger GPU memory space. The networks with deeper architecture do not produce considerable improvements. with the increase of depth,the experiment show that translation quality is decreased on the contrary. Compared with the wider and deeper networks, our model enables diverse encoders and decoders which leads to the translation quality enhancement with less computation overhead.

Although, we carry the experiment on the machine translation task, the ME-MD framework is general and can be applied to other sequence to sequence tasks. The framework is a kind of novel approaches to enhance the neural network performance. Except the implementations that shown in this paper, more categories of encoders and decoders can be introduced into the framework. In the future, we will validate our approach on more language pairs and explore more effective methods to improve the model capacity.

Acknowledgments

Qun Liu's work is partially supported by Science Foundation Ireland in the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Dublin City University funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund. We thank the anonymous reviewers for their insightful comments.

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR2015*, 2015.

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

[Chung *et al.*, 2016] Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of ACL2016*, 2016.

[Dong *et al.*, 2015] Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. Multi-task learning for multiple language translation. In *ACL (1)*, pages 1723–1732, 2015.

[Firat *et al.*, 2016] Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T Yarman Vural, and Yoshua Bengio. Multi-way, multilingual neural machine translation. *Computer Speech & Language*, 2016.

[Jean *et al.*, 2015] Sbastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. On using very large target vocabulary for neural machine translation. In *Proceedings of ACL2014*, volume 1, pages 1–10, 2015.

[Kalchbrenner and Blunsom, 2013] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of EMNLP2013*, pages 1700–1709, Seattle, Washington, USA, October 2013.

[Koehn *et al.*, 2007] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL2007 Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007.

[Luong *et al.*, 2015a] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP2015*, Lisbon, Portugal, September 2015.

[Luong *et al.*, 2015b] Minh Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. *Proceedings of ACL2015*, 27(2):82–86, 2015.

[Luong *et al.*, 2016] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. In *International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2016.

[Meng *et al.*, 2016] Fandong Meng, Zhengdong Lu, Hang Li, and Qun Liu. Interactive attention for neural machine translation. In *Proceedings of COLING2016*, 2016.

[Och and Ney, 2003] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL2002*, pages 311–318. Association for Computational Linguistics, 2002.

[Pascanu *et al.*, 2013] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318, 2013.

[Sennrich *et al.*, 2016] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of ACL2016*, pages 1715–1725, 2016.

[Shen *et al.*, 2016] Shiqi Shen, Yong Cheng, Zhongjun He, Hua Wu, Maosong Sun, and Yang Liu. Minimum risk training for neural machine translation. In *Proceedings of ACL2016*, pages 1683–1692, 2016.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[Stolcke and others, 2002] Andreas Stolcke et al. Srilm-an extensible language modeling toolkit. In *Proceedings of the international conference on spoken language processing*, volume 2, pages 901–904, 2002.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[Tu *et al.*, 2016] Zhaopeng Tu, Zhengdong Lu, yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *Proceedings of ACL*, pages 76–85, 2016.

[Wang *et al.*, 2016] Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. Memory-enhanced decoder for neural machine translation. In *Proceedings of EMNLP2016*, 2016.

[Wu *et al.*, 2016] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[Zeiler, 2012] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[Zhou *et al.*, 2016] Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. Deep recurrent models with fast-forward connections for neural machine translation. In *Proceedings of EMNLP2016*, 2016.

[Zoph and Knight, 2016] Barret Zoph and Kevin Knight. Multi-source neural translation. In *Proceedings of NAACL-HLT*, pages 30–34, 2016.