Neural Machine Translation with Decoding-History Enhanced Attention

Mingxuan Wang[1] **Jun Xie**[1] **Zhixing Tan**[1] **Jinsong Su**[2] **Deyi Xiong**[3] **Chao bian**[1]
[1]Mobile Internet Group, Tencent Technology Co., Ltd
{xuanswang,stiffxie,zhixingtan,chaobian}@tencent.com
[2]Xiamen University, Xiamen, China
{jssu}@xmu.edu.cn
[3]Soochow University, Suzhou, China
{dyxiong}@suda.edu.cn

Abstract

Neural Machine Translation (NMT) with source side attention have achieved remarkable performance. however, there has been little work exploring to attend to the target side which can potentially enhance the memory capbility of NMT. We reformulate a *Decoding-History Enhanced Attention mechanism* (DHEA) to render NMT model better at selecting both source side and target side information. DHEA enables a dynamic control on the ratios at which source and target contexts contribute to the generation of target words, offering a way to weakly induce structure relations among both source and target tokens. It also allows training errors to be directly back-propagated through short-cut connections and effectively alleviates the gradient vanishing problem. The empirical study on Chinese-English translation shows that our model with proper configuration can improve by 0.9 BLEU upon Transformer and achieve the best reported results in the same dataset. On WMT14 English-German task and a larger WMT14 English-French task, our model achieves comparable results with the state-of-the-art NMT systems.

1 Introduction

Neural Machine Translation(Sutskever et al., 2014) generally adopts an encoder-decoder framework, where the encoder reads and encodes the input text into a distributed representation and the decoder generates translated text conditioned on the input representation. A potential issue with this encoder-decoder approach is that a neural network needs to be able to compress all the necessary information of a source sentence into a fixed-length vector. This may make it difficult for the neural network to cope with translating long sentences. A recent successful extension of NMT models is the attention mechanism which conducts a soft search over source tokens and yields an *attentive vector* to represent the most relevant segments of the source sentence for the current decoding state (Bahdanau et al., 2014; Jean et al., 2015; Luong et al., 2014; Vaswani et al., 2017).

The typical attention mechanism frees the neural translation model from having to squash all the information of the source sentence into a fixed vector, however, it ignores some important information hidden in the target sequence (Cheng et al., 2016). At least three challenges still remains in the translation process. The first issue relates to the memory compression problems in the decoding process. During each translation step, a hidden state vector implicitly maintains at least two types of information, including both the most relevant source contexts and the language model over the partially translated sentence. The memory capacity of a single dense vector is not powerful enough to store these information. The second challenge comes with the training and optimization problem associated with vanishing or exploding gradients, which has been studied in the context of vanilla RNNs. Although some RNN variants such as LSTMs provide a temporal shortcut path to avoid the problem in the temporal domain, it is not guaranteed to be sufficient. At last, it should be acknowledged that the target side context is solely based on the sequence model which, in practice, is prone to a fixed vector and lacks the ability to capture effectively nonsequential dependencies among words.

Proceedings of the 27th International Conference on Computational Linguistics, pages 1464–1473
Santa Fe, New Mexico, USA, August 20-26, 2018.

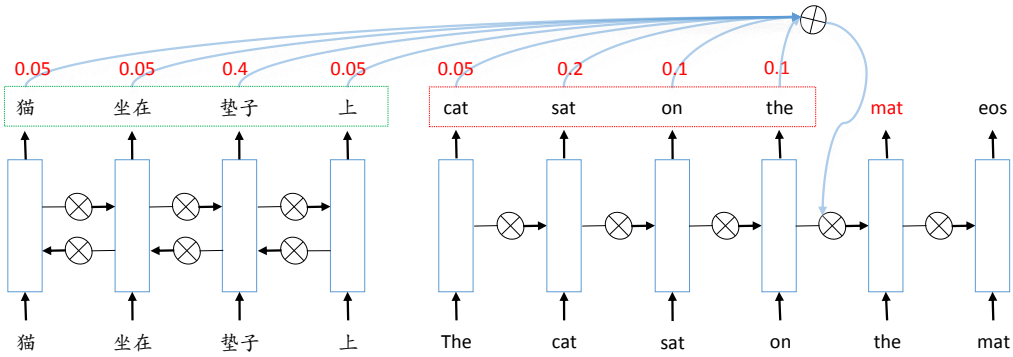

Figure 1: DHEA: *Decoding-History Enhanced Attention mechanism.* The interaction with hidden states that cross both the source and target side can be enhanced by these short-cut connections. The attention weights controls the individual contribution of each hidden states. In this example, the semantic dependency between the word "sat" and "mat" can be be explicitly captured.

In order to address these limitations, we introduce a *Decoding History Enhanced Attention* (DHEA) as a powerful extension to the typical attention based NMT architecture, where the attention over decoding history contributes directly to the prediction of the next word. As illustrated in Figure 1, DHEA explicitly looks back at multiple preceding steps and automatically decides how much previous information should be "seen" by weighting them. Basically, this sort of attention paid on the decoding history facilitates the flow of information from the distant past and is able to emphasize any of the previously translated words, hence it enables the learning of syntactic structures which are useful for the translation task. It also introduces some shortcut connections which connect far-away states to ensure training error signal to be back-propagated directly and alleviate gradient vanishing problem. Further more, we apply attention on the source and target side contexts synchronously, which can better characterize the dependencies within the two sides. Acting in this way, the system learns to determine the ratos of information from the two sides and properly select the amount of context information with regard to the generation of next word in the decoding process. For example, content words in the target sentence should depend more on the source context but less on its prefix (*e.g.* "*mat*" depends more one "*dianzi*" and takes "*cat*" as complementarity). In contrast, function words in the target sentence are often more related to the target context (*e.g.* "*on*" after "*sat*").

Our evaluation on three language pairs shows that the proposed model improves over several baselines, with only a small increase in computational overhead. On the NIST Chinese-English task, DHEA with proper settings yields the best reported result and a 0.9 BLEU improvement over a strong NMT system (Transformer). On WMT English-German and English-French tasks, it again leads to consistent improvements and achieves performance superior or comparable to the state-of-the-art. Finally, we analyze the learned attention function, providing additional insights to its actual contributions.

2 Neural Machine Translation

The NMT model aims to compute the conditional distribution of generating a sentence in a target language given a sentence in a source language, denoted by $p_\theta(y|x)$, where $x = \{x_1 \cdots, x_T\}$ and $y = \{y_1, \cdots, y_{T'}\}$ represent the source and target sentences as sequences of words respectively. θ is a set of parameters usually trained to maximize the conditional log-probability of the correct translation y given the source sentence x:

$$\theta^* = \arg\max_\theta \sum_{i=1}^{N} \log p(y^{(i)}|x^{(i)}) \quad (1)$$

where N is the size of the training corpus. In this section, we will first briefly introduce the NMT model used in our work. Our model modifies the attention based architecture proposed by Bahdanau et

al. (2014), and implements as a deep stack LSTM framework. The whole architecture has three main components: encoder, decoder and attention mechanism.

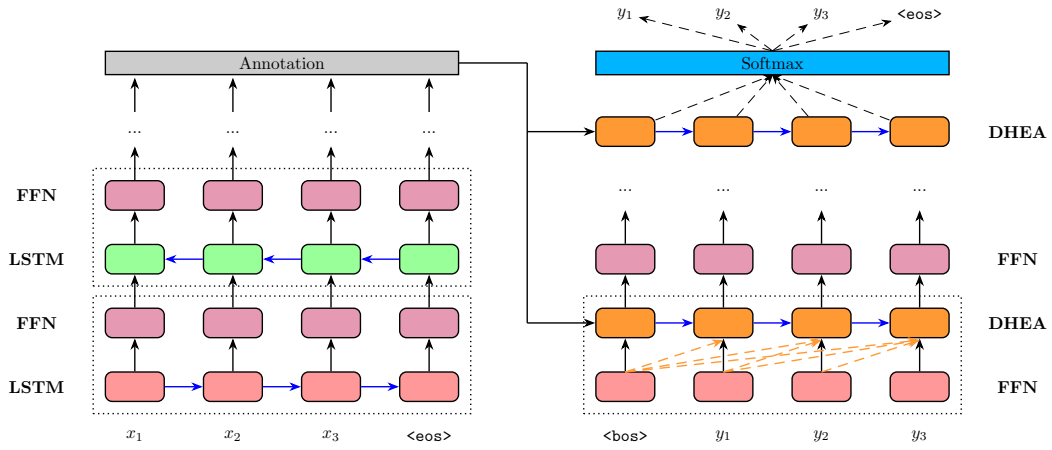

Figure 2: The structure of deep neural machine translation model with DHEA. The dashed arrow directly connects the decoding history and the current state. If we remove this connection, the model degrades to the source side attention based NMT.

Encoder The goal of the encoder is to build meaningful representations of source sentences. The typical encoder consists of a bidirectional RNN which processes the raw input in backward and forward direction with two separate layers, and then concatenates them together. In this work, we choose another bidirectional approach to process the sequence in order to learn more temporal dependencies. Specifically, an RNN layer processes the input sequence in a forward direction. The output of this layer is taken by an upper RNN layer as input, processed in a reverse direction (Zhou et al., 2016).

More formally,The encoder network reads the source input $x = \{x_1, ..., x_T\}$ and processes it into a source side memory $M^{\mathrm{s}} = \{h_1, h_2 \cdots, h_T\}$. where $x_i \in \mathbb{R}^{d_x}$. The output on layer ℓ is

$$h_t^\ell = \begin{cases} x_t, & \ell = 1 \\ \mathrm{LSTM}(h_{t+d}^\ell, h_t^{\ell-1}), & \ell > 1 \end{cases} \tag{2}$$

where

- $h_t^\ell \in \mathbb{R}^{d_h}$ gives the output of layer ℓ at location t.
- The directions are marked by a direction term $d = (-1)^\ell$. If we fixed d to -1, the input will be processed in forward direction, otherwise backward direction.
- We only apply the top-most hidden states as the source side memory which is then fed to the decoder.

Decoder The decoder uses another RNN to generate the translation $y = \{y_1, \cdots, y_{T'}\}$ based on the source side memory M^{s} produced by the encoder. In this work, we use a unidirectional deep stacked LSTM for the decoder RNN. More formally, the hidden state $s_t^\ell \in \mathbb{R}^{d_s}$ on decoder layer ℓ is

$$s_t^\ell = \begin{cases} y_t, & \ell = 1 \\ \mathrm{LSTM}(s_{t-1}^\ell, s_t^{\ell-1}, c_t^\ell), & \ell > 1 \end{cases} \tag{3}$$

Where y_t is the target word embedding at time step t and the context vector $c_t^\ell \in \mathbb{R}^{d_h}$ is dynamically obtained from M^{s} by the attention mechanism. At inference stage, we only utilize the top-most hidden states s to make the final prediction with a softmax layer:

$$p(y_i | y_{<i}, x) = \mathrm{softmax}(s_i W_o) \tag{4}$$

Attention The attention mechanism allows the decoder to select which parts of the source sentence are more useful to predict the next output word. It can be viewed as mapping a query and a set of key-value pairs to an output. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. More specifically, the attention model get context vector c_t after reading from the source representation M^{src}:

$$\begin{aligned} c_i^\ell &= \text{Read}(s_i, M^{\text{s}}) \\ &= \sum_{j=1}^{T} \alpha_{ij}^\ell (h_j W_V^\ell) \end{aligned} \tag{5}$$

Each weight coefficient, α_{ij}^ℓ , is computed using a softmax function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \tag{6}$$

And e_{ij}^ℓ is computed using a compatibility function that compares two input elements. The two most commonly used attention functions are additive attention, and dot-product attention. Additive attention computes the weights of the values using a feed-forward network with a single hidden layer. In this work, we consider dot-product with a scaling factor as the compatibility function for its simplicity and efficiency:

$$e_{ij}^\ell = \frac{1}{\sqrt{d_s}} (s_i^{\ell-1} W_Q^\ell)(h_j W_K^\ell)^T \tag{7}$$

Linear transformations of the inputs add sufficient expressive power. $W_K, W_V \in \mathbb{R}^{d_h \times d_s}$ and $W_Q \in \mathbb{R}^{d_s \times d_s}$ are parameter matrices. In general d_s and d_h are set to the same. These parameter matrices are unique per layer. It should be noticed that in Eqn.(7), we apply $s_i^{\ell-1}$ to compute the weights of layer ℓ on the values. In this way the query vector can be prepared in anvance which benifits the implementation of using highly optimized matrix multiplication code.

3 Decoding-History Enhanced Attention for NMT

From Eqn.(5), it is not difficult to observe that the attention weights are associated with the only source side memory. As we have argued before, it is easy for the conventional attention-based NMT models to suffer from generating incoherent texts due to the conflict between the source side attention mechanism and the decoding history. We therefore propose decoding-history enhanced attention to better control over target side contexts. In addition to the source memory M^{s}, DHEA is equipped with a buffer memory M^B as an extension to the conventional approach. M^B is identical to the source side memory M^{S}, except that M^B changes per time step. M_i^B is defined as $\{s_1, s_2, \cdots, s_i\}$. The new context vector is then defined as a combination of the two sides contexts:

$$\hat{c} = \text{Read}(s, M^{\text{S}}, M^{\text{B}}) \tag{8}$$

In this work investigate three strategies for integrating these contexts.

Sum Combination At time step i, DHEA reads from M^B:

$$\begin{aligned} z_i^\ell &= \text{Read}(s_i, M^{\text{B}}) \\ &= \sum_{j=1}^{i} \alpha_{ij}^\ell (s_j^{\ell-1} W_V^\ell) \end{aligned} \tag{9}$$

Similarly, α_{ij} is computed by Eqn.(6) where $j \leq i$ and e_{ij}^ℓ is computed as :

$$e_{ij}^\ell = \frac{1}{\sqrt{d_s}} (s_i^{\ell-1} W_Q^\ell)(s_j^{\ell-1} W_K^\ell)^T \tag{10}$$

One simple way to aggregate information from z_i and c_i is by summing them, then the new context vector is computed as:

$$\hat{c} = z + c \tag{11}$$

It is also worth mentioning that we use $s^{\ell-1}$ as the context information to update the hidden state on layer ℓ, since the lower layer states can be perpared in advance to facilitate parallel training.

Gate Combination As argued by Tu et al. (2016a), the source side context and the target side context plays a different role during the decoding process, we therefore design a context gate which assigns an element-wise weight to the two-side input:

$$\hat{c} = g(c, z) \cdot c + (1 - g(c, z)) \cdot z \tag{12}$$

where $g(c, z) \in (0, 1)$ is a sigmoid neural network which dynamically controls the amount of information flowing from the source and target contexts. $\hat{c}$ is the new context vector fed to the decoder in Eqn.(3) which refines $s_t^\ell = \text{LSTM}(s_{t-1}^\ell, s_t^{\ell-1}, \hat{c}_t^\ell)$.

With the gated control, the new context vector $\hat{c}$ can be rectified based on the decoding history and the source side information. The memory storing useful information of the partial translation can encourage the model to translate contents that are less repeated compared with the already translated contents.

Hybird Combination Considering that we apply the similar attention function to read from the sourc-side memory and the target side memory, it is natural to merge these two memorys into a hybird memory block $M^{\text{S+B}}$. After that, the decoder directly reads from this memory block.

$$\hat{c} = \text{Read}(s, M^{\text{S+B}}) \tag{13}$$

Where the memory block $M^{\text{S+B}}$ is the concatenation of M^{S} and M^{B}. The method can be viewd as a simple implementation of gate combination where the the gating weights are equal to the attention weights.

4 Experiments

4.1 Datasets

We mainly evaluated our approaches on the widely used NIST Chinese-English translation task. In order to show the usefulness of our approaches, we also provide results on other two translation tasks: English-French, English-German. The evaluation metric is BLEU. For Chinese-English task, we apply case-insensitive NIST BLEU. For other tasks, we tokenized the reference and evaluated the performance with multi-bleu.pl. The metrics are exactly the same as in previous work (Papineni et al., 2002).

For Chinese-English, our training data consists of 1.25M sentence pairs extracted from LDC corpora[1], with 27.9M Chinese words and 34.5M English words respectively. We choose NIST 2002 (MT02) dataset as our development set, and the NIST 2003 (MT03), 2004 (MT04), 2005 (MT05), and 2006 (MT06) datasets as our test sets.

For English-German, to compare with the results reported by previous work, we used the same subset of the WMT 2014 training corpus that contains 4.5M sentence pairs with 91M English words and 87M German words. The concatenation of news-test 2012 and news-test 2013 is used as the validation set and news-test 2014 as the test set.

To evaluate at scale, we also report the results of English-French. To compare with the results reported by previous work on end-to-end NMT, we used the same subset of the WMT 2014 training corpus that contains 36M sentence pairs. The concatenation of news-test 2012 and news-test 2013 serves as the validation set and news-test 2014 as the test set.

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

4.2 Training details

Our training procedure and hyper parameter choices are similar to those used by (Vaswani et al., 2017). In more details, we limited the source and target vocabularies to the most frequent $30K$ words in Chinese-English translation. For English-German translation and English-French translation, we use $50K$ sub-word tokens as vocabulary based on Byte Pair Encoding(Sennrich et al., 2015).

We initialized parameters by sampling each element from the Gaussian distribution with mean 0 and variance 0.01^2. Parameter optimization was performed using stochastic gradient descent. Adam (Kingma and Ba, 2015) was used to automatically adapt the learning rate of each parameter ($\beta_1 = 0.9, \beta_2 = 0.98$ and $\epsilon = 10^{-8}$). To avoid gradient explosion, the gradients of the cost function which had ℓ_2 norm larger than a predefined threshold 25 were normalized to the threshold (Pascanu et al., 2013). We batched sentence pairs by approximate length, and limited input and output tokens per batch to 8192 per GPU. Each resulting training batch contained approximately 60,000 source and 60,000 target tokens. We trained our NMT model with the sentences of length up to 150 words in the training data. During training, we employed label smoothing of value $\epsilon = 0.1$(**?**).

Translations were generated by a beam search and log-likelihood scores were normalized by the sentence length. We used a beam width of 8 and length penalty $\alpha = 0.6$ in all the experiments. Dropout was applied on each layer to avoid over-fitting (Hinton et al., 2012). The dropout rate was set to 0.1. Except when otherwise mentioned, NMT systems had 6 layers encoders and 4 layers decoders. We trained for 300,000 steps on 8 M40 GPUs, and averaged the last 20 checkpoints, saved at 30 minute intervals. For our small model, the dimensions of all the hidden states were set to 512 and for the big model, the dimensions were set to 1024.

4.3 Results on Chinese-English Translation

SYSTEM	MT03	MT04	MT05	MT06	AVE.
Existing systems					
Moses	31.61	33.48	30.75	30.85	31.67
DEEPLAU(Wang et al., 2017)	39.35	41.15	38.07	37.29	38.97
FRNN+PRNN (Zheng et al., 2017)	37.90	40.37	36.75	34.55	37.39
COVERAGE+Context Gate(Tu et al., 2016a)	-	-	34.13	34.83	-
Transformer (Vaswani et al., 2017)	45.16	46.81	44.62	43.53	45.03
Our deep NMT systems					
Source Attn.	45.08	46.90	44.32	43.13	44.85
DHEA + Hybird Comb.	46.58	47.20	45.45	43.47	45.66
DHEA + Sum Comb.	46.38	47.15	45.30	43.60	45.50
DHEA + Gate Comb.	46.60	47.73	45.35	43.97	**45.90**

Table 1: Case-insensitive BLEU scores on Chinese-English translation.

Table 1 shows BLEU scores on Chinese-English datasets. Clearly DHEA leads to remarkable improvements over their competitors. Compared to the source side attention based model, DHEA is $+1.05$ BLEU score higher on average four test sets, showing the modeling power gained from the decoding history. Among the combination method of the contexts, the context gate function drives consistent improvements over sum and hybrid function by $+0.4$ and $+0.2$ in terms of BLEU. We conjecture it is because the adaptive gate function conditioned on the decoding history make it able to automatically decide how much information should be transferred during decoding to the next step.

To show the power of DHEA, we also make a comparison with previous work. Our best single model outperforms both a phrased-based MT system (Moses) as well as a strong source attention-based NMT system (DeepLau) by $+14.2$ and $+6.8$ BLEU points respectively on average. The result is also better than some other state-of-the-art variants of attention-based NMT model with big margins. Compared to Transformer, DHEA yields a gain of $+0.9$ BLEU and achieves the best performance ever reported this dataset.

4.4 Results on English-German and English-French Translation

SYSTEM	Architecture	EN-Fr BLEU	EN-DE BLEU
Existing systems			
Buck et al. (2014)	Winning WMT14 system	35.7	20.7
Wu et al. (2016)	GNMT + Ensemble	40.4	26.3
Gehring et al. (2017)	ConvS2S	40.51	25.16
Vaswani et al. (2017)	Transformer (small)	38.1	27.3
Vaswani et al. (2017)	Transformer (large)	41.0	28.4
Our deep NMT systems			
this work	Source Attn (small)	39.3	27.5
this work	DHEA + Gate Comb. (small)	40.4	27.9
this work	DHEA + Gate Comb. (large)	**41.6**	**28.7**

Table 2: Case-sensitive BLEU scores on English-German and English-French translation. Our proposed model outperforms the state-of-the-art models including the Transformer (Vaswani et al., 2017).

The results on English-German and English-French translation are presented in Table 2. We compare our NMT systems with various other systems including the winning system in WMT14 (Buck et al., 2014), a phrase-based system whose language models were trained on a huge monolingual text, the Common Crawl corpus. For end-to-end NMT systems, to the best of our knowledge, GNMT is the best RNN based NMT system. Transformer (Vaswani et al., 2017) is currently the SOTA system which is about 2 BLEU points better than GNMT on the English-German task and 0.6 BLEU points better than GNMT on the English-French task.

DHEA achieves a 0.6 BLEU score improvement over the state-of-the-art on the English-to-German task for the smaller network and 0.3 BLEU improvement for the larger network. In the case of the larger English-to-French task, we obtain a 2.3 BLEU improvement for the smaller model and a 0.6 improvement for the larger one. Also, note that the performance of the smaller model for DHEA is close to that of the larger baseline model, especially for the English-to-French task. This suggests that DHEA better utilizes available model capacity.

4.5 Analysis

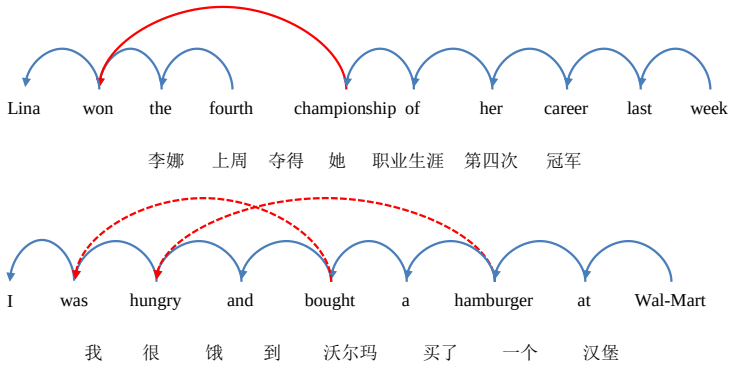

Figure 3: Examples of the target side attention during decoding process. Solid arrows denote which word is being focused when attention is computed, but not the direction of the relation. Red lines indicate long-term dependencies. Dashed lines denote which word is bing second focused.

In order to better understand whether the network benefits from the decoding history, we biefly visualized the skeleton of the target side attention. In Figure 3, although we explicitly allow DHEA to attend to any memory cell, much attention selects the immediately previous word. This agrees with the linguistic intuition that long-term dependencies are relatively rare and hard to learn. This model still finds some

long-term dependencies among words (e.g., the dependency between *won* and *championship*, *bought* and *was*, *hungry* and *hamburger*). These detected dependencies reflect some head-modifier relations in dependency graphs.

5 Related Work

Memory Networks There are variety of studies proposed to increase the LSTM memory capacity by using memory networks. The two most salient examples are Neural Turing Machine (NTM) (Graves et al., 2014) and Memory Network (Weston et al., 2014). Cheng et al. (2016) propose a machine reading simulator which processes text incrementally from left to right. In the NMT task, Wang et al. (2016) present a decoder enhanced decoder with an external shared memory which extends the capacity of the network and has the potential to read, write, and forget information. In fact DHEA can be viewed as a special case of memory networks, with only reading mechanism for the translation task. Quite remarkably DHEA incorporates two different types of memory (source memory and decoding history memory) and significantly improves upon state-of-the-arts.

Attention Mechanism Attention in neural networks (Bahdanau et al., 2014; Luong et al., 2015) is designed to assign weights to different inputs instead of threating all input sequences equally as original neural networks do. A number of efforts have been made to improve the attention mechanism(Tu et al., 2016b; Mi et al., 2016; Zhang et al., 2017). Some of them incorporated the previous attention history into the current attention for better alignment, but none of them are based on the decoding history.

The application of self-attention mechanisms in RNNs have been previously studied, and in general, it appears to capture syntactic dependencies among distant words (Liu and Lapata, 2017; Lee et al., 2017; Kim et al., 2017; Lin et al., 2017). Vaswani et al. (2017) resort to self-attention mechanism and showed outstanding performance. Our approach is diffrent from their work in two aspect. First, our method can be viewed as a variant of RNN decoder which allows a form of memory, thus has the the potential to better handle sentences of arbitrary length. Second, we forcus on controlling the information flow between the source side memory and the target side memory and design a gate to balance the contribution of the two-sides.

Recurrent Residual Networks Our work is also related to residual connections, which have been shown to improve the learning process of deep neural networks by addressing the vanishing gradient problem (He et al., 2015; Szegedy et al., 2016). Recently, several architectures using residual connections with LSTMs have been proposed (Kim et al., 2017; Wang, 2017; Wang and Tian, 2016) for sequence prediction. These connections create a direct path from previous layers, helping the transmission of information. Related to our work, Miculicich et al. (2018) propose a target sideattentive residual recurrent network for decoding, where attention over previous words contributes directly to the prediction of the next word. Comparatively, DHEA attends to the previous hidden state and make a combination with the source context.

Exploiting Contextual Information A thread of work in sequence to sequence learning attempts to exploit auxiliary context information(Wang and Cho, 2016; Li et al., 2017; Zhang and Zong, 2016). Recently Tu et al. (2016a) propose using context gates in NMT to dynamically control the contributions from the source contexts and the RNN hidden state. Our approach focuses on integrating the decoding history and the source side context to NMT architecture. In addition, we have a multi-layer approach to better utilize the contextual information. Experiments in Section 4.3 show the superiority of DHEA.

In the same period of our work, Lin et al. (2018) and Xia et al. (2017) first turn eyes to the target side attention of NMT architecture. Our approach share the similar idea with these work. The diffrence lies in that we concernd more about the integrating of the souce side context and the target side context and designed three types of combination functions. In addition, we approached in a multi-layer way which is more effective.

6 Conclusion

We presented a decoder history enhanced attention mechanism to enrich the target side contextual information for neural machine translation. Our empirical study in three translation tasks shows that it can significantly improve the performance of NMT.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Christian Buck, Kenneth Heafield, and Bas Van Ooyen. 2014. N-gram counts and language models from the common crawl. In *LREC*, volume 2, page 4. Citeseer.

Jianpeng Cheng, Li Dong, and Mirella Lapata. 2016. Long short-term memory-networks for machine reading. *empirical methods in natural language processing*, pages 551–561.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China, July. Association for Computational Linguistics.

Jaeyoung Kim, Mostafa El-Khamy, and Jungwon Lee. 2017. Residual LSTM: design of a deep recurrent architecture for distant speech recognition. *CoRR*, abs/1701.03360.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. *international conference on learning representations*.

Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2017. Recurrent additive networks. *CoRR*, abs/1705.07393.

Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. 2017. Modeling source syntax for neural machine translation. pages 688–697.

Zhouhan Lin, Minwei Feng, Cicero Nogueira Dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding.

Junyang Lin, Shuming Ma, Qi Su, and Xu Sun. 2018. Decoding-history-based adaptive control of attention for neural machine translation.

Yang Liu and Mirella Lapata. 2017. Learning structured text representations. *CoRR*, abs/1705.09207.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

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

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation.

Lesly Miculicich Werlen, Nikolaos Pappas, Dhananjay Ram, and Andrei Popescu-Belis. 2018. Self-attentive residual decoder for neural machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2013. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

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

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*.

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2016a. Context gates for neural machine translation. *CoRR*, abs/1608.06043.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016b. Modeling coverage for neural machine translation. *ArXiv eprints, January*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Tian Wang and Kyunghyun Cho. 2016. Larger-context language modelling with recurrent neural network. In *Meeting of the Association for Computational Linguistics*, pages 1319–1329.

Yiren Wang and Fei Tian. 2016. Recurrent residual learning for sequence classification. In *Conference on Empirical Methods in Natural Language Processing*, pages 938–943.

Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2016. Memory-enhanced decoder for neural machine translation. *empirical methods in natural language processing*, pages 278–286.

Mingxuan Wang, Zhengdong Lu, Jie Zhou, and Qun Liu. 2017. Deep neural machine translation with linear associative unit. *CoRR*, abs/1705.00861.

Cheng Wang. 2017. RRA: recurrent residual attention for sequence learning. *CoRR*, abs/1709.03714.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *CoRR*, abs/1410.3916.

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

Yingce Xia, Fei Tian, Tao Qin, Nenghai Yu, and Tie Yan Liu. 2017. *Sequence Generation with Target Attention*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of EMNLP*.

Jinchao Zhang, Mingxuan Wang, Qun Liu, and Jie Zhou. 2017. Incorporating word reordering knowledge into attention-based neural machine translation. In *Meeting of the Association for Computational Linguistics*, pages 1524–1534.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Lili Mou, Xinyu Dai, Jiajun Chen, and Zhaopeng Tu. 2017. Modeling past and future for neural machine translation. *CoRR*, abs/1711.09502.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *arXiv preprint arXiv:1606.04199*.