A Context-Aware Recurrent Encoder for Neural Machine Translation

Biao Zhang, Deyi Xiong, Jinsong Su, and Hong Duan

***Abstract*—Neural machine translation (NMT) heavily relies on its encoder to capture the underlying meaning of a source sentence so as to generate a faithful translation. However, most NMT encoders are built upon either unidirectional or bidirectional recurrent neural networks, which either do not deal with future context or simply concatenate the history and future context to form context-dependent word representations, implicitly assuming the independence of the two types of contextual information. In this paper, we propose a novel context-aware recurrent encoder (CAEncoder), as an alternative to the widely-used bidirectional encoder, such that the future and history contexts can be fully incorporated into the learned source representations. Our CAEncoder involves a two-level hierarchy: The bottom level summarizes the history information, whereas the upper level assembles the summarized history and future context into source representations. Additionally, CAEncoder is as efficient as the bidirectional RNN encoder in terms of both training and decoding. Experiments on both Chinese–English and English–German translation tasks show that CAEncoder achieves significant improvements over the bidirectional RNN encoder on a widely-used NMT system.[1]**

***Index Terms*—Context-aware encoder, neural machine translation (NMT), natural language processing, recurrent encoder.**

I. Introduction

With the rapid development of deep neural models, we have witnessed that research interests in the machine translation literature have been shifting from statistical machine translation (SMT) to end-to-end neural machine translation (NMT) [1]–[5]. NMT is a large, single neural network that consists of two parts: an *encoder* encodes a source sentence into its semantic representations, from which a *decoder* generates the corresponding target translation word by word [3]. To ensure the faithfulness of the translation, the encoder has to be capable of encoding all necessary information of the source sentence into its semantic representations.

Most NMT systems use encoders based on recurrent neural network (RNN). A recurrent encoder takes as input a variable-length sequence and sequentially encode a token from left to right, or in an inverse order. Although several efficient variants have been proposed, e.g., GRU [6] and LSTM [7], there are inherent shortcomings inside these RNN encoders. For example, suppose a left-to-right RNN encoder is reading the following sentence until the word "*bank*":

The boy leapt from the bank into the cold water. (1)

Without the important future context "*cold water*", the RNN encoder would not know whether the word "*bank*" refers to *the edge of a river* or *a financial institution*. To successfully induce the appropriate semantic representation of "*bank*", we need to incorporate the future context into encoder.

One way to solve this problem is to stack multiple RNNs such that the lower RNNs are able to provide full contextual information for the higher RNNs [1], [8]. However, training deep recurrent models needs batches of optimization tricks which are nontrivial. Yet another solution is to use bidirectional RNNs so that one RNN encodes the forward history and the other the backward history. Although bidirectional RNNs perform well in practice, they are relatively weak in assembling the forward and backward representations. The typical concatenation operation used in bidirectional RNNs implicitly assumes that both forward and backward representations are independent of each other, which, on the other hand, increases the burden of the decoder since this operation doubles the dimensionality of the RNN representations with extra computational costs. We, instead, prefer a novel encoder that 1) naturally deals with the history and future contexts and 2) is quite efficient in computation and training.

In this paper, we propose a context-aware recurrent encoder (CAEncoder) for NMT system. The intuition behind is that we can compute the future context representations in advance and then feed them into the RNN model. Therefore, the left questions are 1) how do we learn the future context representations? and 2) how are these precomputed future context representations integrated with the history context representations? For the first question, we still use RNN to induce the future context representations on condition that the used RNN processes the input sequence following a different direction from the CAEncoder. To answer the second question, we design a two-level

Manuscript received April 9, 2017; revised July 19, 2017 and August 18, 2017; accepted September 7, 2017. Date of publication September 11, 2017; date of current version November 27, 2017. This work was supported in part by the National Natural Science Foundation of China under Grants 61672440, 61622209, 61573294, and 61403269, in part by the Scientific Research Project of National Language Committee of China under Grant YB135-49, and in part by the Natural Science Foundation of Fujian Province under Grant 2016J05161. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Imed Zitouni. *(Corresponding author: Jinsong Su.)*

B. Zhang, J. Su, and H. Duan are with the Software School, Xiamen University, Xiamen 361005, China (e-mail: zb@stu.xmu.edu.cn; jssu@xmu.edu.cn; hudan@xmu.edu.cn).

D. Xiong is with the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, China (e-mail: dyxiong@suda.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2017.2751420

[1]Source code will be available at https://github.com/DeepLearnXMU/CAEncoder-NMT.

2329-9290 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

hierarchy: the bottom level processes the original tokens to summarize what has been read so far, while the upper level assembles this summarization and the precomputed future context to generate the semantic representation for a source word. Let's revisit the above example. When reading the word "*bank*", CAEncoder already knows the history and future context "*leapt, cold water*" so that it is able to induce the correct meaning *the edge of a river*.

We train our CAEncoder-enhanced NMT system with the same optimization algorithm as the baseline NMT system. Our CAEncoder can be regarded as an alternative to the widely used bidirectional RNN encoder. We conduct a series of experiments on Chinese-English translation tasks. The results show that our model significantly outperforms the widely-used NMT system by around 2.0 BLEU points. We also verify our model on the more challenging WMT English-German translation tasks, achieving very encouraging performance. Further analysis on translations reveals its capacity in appropriately modeling the semantics of source sentences.

II. Related Work

NMT systems heavily rely on a semantic encoder to encode source sentences into real-valued semantic representations [1], [3]. To enhance the capacity of NMT encoder, Sutskever *et al.* [3] adopts a multi-layered LSTM encoder. Zhou *et al.* [8] design a fast-forward connection to enable flexible training for deep encoders; Meng *et al.* [9] propose a series of reading and writing memory components to model complicated relations between sequences necessary for translation. Google NMT system also uses a multi-layered recurrent encoder [10]. Although these deep NMT systems are very powerful, training such deep models is nontrivial. Instead of developing deep neural encoders, Eriguchi *et al.* [11] build a tree-based encoder to incorporate the syntactic structure inside a source sentence explicitly. Su *et al.* [12] propose a lattice-based encoder to handle the tokenization ambiguity. The former syntactic encoder relies on the accuracy of parse trees, and the latter encoder needs different word segmentation results. The problems with the two encoders are their low robustness and high computational cost. On the contrary, our encoder is very simple, efficient and robust.

Additionally, there are some recent attempts that explore different encoder architectures beyond the recurrent architecture. For example, Kalchbrenner *et al.* [13] try to perform machine translation with neural networks in linear time and propose a convolutional neural encoder. Gehring *et al.* [14] employ a succession of convolutional layers to encode source sentences. In contrast, our work still uses the recurrent architecture due to its effectiveness in modeling sequences.

Our encoder is a variant of RNN. RNN takes as input a variable-length sequence of tokens, and processes each token recursively while maintaining an internal hidden state. Typically, the hidden state is able to encode context information from history tokens. However, conventional RNNs often suffer from the vanishing and the exploding gradient problems during training [15], which results in difficulties in capturing long-term dependencies. To solve this problem, Hochreiter and Schmidhuber [7] propose the well-known Long Short-Term Memory (LSTM) which explicitly introduces a memory cell to memorize long-term dependencies together with input, forget and output gates for information controlling. Considering the inefficiency of LSTM in computation, Chung *et al.* [6] further simplify this architecture, and introduce the gated recurrent unit (GRU) where only a reset and an update gate remains. Both LSTM and GRU have been demonstrated efficient in modeling long-term dependencies through several empirical evaluations [1], [3], [6]. We design our encoder based on these methods, especially the GRU model.

Our work is also related to that of Choi *et al.* [16] who contextualize word embeddings using a nonlinear bag-of-words representation of the source sentence. Our work differs significantly from theirs since we focus on how to better encode source sentences upon vanilla word embeddings while they focus on how to disambiguate word embeddings before the vanilla encoding.

III. Neural Machine Translation

We briefly review NMT in this section, especially the attention-based NMT. Given a collection of source and target sentence pairs $(\mathbf{x}, \mathbf{y})$, NMT estimates the conditional probability $p(\mathbf{y}|\mathbf{x})$ via directly mapping the source sentence $\mathbf{x} = \{x_1, \ldots, x_n\}$ to its target translation $\mathbf{y} = \{y_1, \ldots, y_m\}$:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{m} p(y_j|\mathbf{x}, \mathbf{y}_{<j}) \tag{1}$$

where $\mathbf{y}_{<j} = \{y_1, \ldots, y_{j-1}\}$ is a partial translation. Similar to a neural language model, NMT predicts the j-th target word using a neural network:

$$p(y_j|\mathbf{x}, \mathbf{y}_{<j}) = softmax\left(g(E_{y_{j-1}}, \mathbf{s}_j, \mathbf{c}_j)\right) \tag{2}$$

$E_{y_{j-1}} \in \mathbb{R}^{d_w}$ is the embedding of previously generated target word y_{j-1}, $\mathbf{s}_j \in \mathbb{R}^{d_h}$ is the j-th target-side hidden state, $\mathbf{c}_j \in \mathbb{R}^{d_h}$ is the translation-sensitive semantic vector and $g(\cdot)$ is a highly non-linear function. Readers could refer to [3] for more details. Here, we mainly highlight the attention mechanism and encoder.

Attention Mechanism This mechanism aims at generating $\mathbf{c}_j$ according to the previous target-side hidden state $\mathbf{s}_{j-1}$ and the encoded source word representations $\mathbf{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_n\}$. To achieve this, it automatically pays attention to the parts of the source sentence that are relevant to predict the next target word:

$$\mathbf{c}_j = \sum_i \alpha_{ji} \mathbf{h}_i \tag{3}$$

where α_{ji} measure the degree of relevance of a source word x_i for predicting the target word y_j. Typically, we treat it as a soft word alignment which can be directly visualized. We compute this attention weight α_{ji} using another neural model:

$$\alpha_{ji} = \frac{exp\left(a(\mathbf{s}_{j-1}, \mathbf{h}_i)\right)}{\sum_i exp\left(a(\mathbf{s}_{j-1}, \mathbf{h}_i)\right)} \tag{4}$$

where $a(\cdot)$ is an *alignment model* as in [3].

The generated $\mathbf{c}_j$ identifies which source words are responsible for the current target word prediction. Its computation is based on well-encoded source representations $\mathbf{h}_i$.

Encoder NMT usually employs a bidirectional RNN as an encoder, where a forward RNN reads the source sentence from left to right while a backward RNN operates in an opposite direction:

$$\begin{aligned}\overrightarrow{h}_i &= f_{enc}(\overrightarrow{h}_{i-1}, E_{x_i}) \\ \overleftarrow{h}_i &= f_{enc}(\overleftarrow{h}_{i+1}, E_{x_i})\end{aligned} \tag{5}$$

$E_{x_i} \in \mathbb{R}^{d_w}$ is the embedding of source word x_i, and $\overrightarrow{h}_i, \overleftarrow{h}_i \in \mathbb{R}^{d_h}$ are the hidden states generated in two directions. We employ GRU for the encoding function $f_{enc}(\cdot)$. The source representation is obtained by concatenating the forward and backward hidden states, $\mathbf{h}_i = [\overrightarrow{h}_i^T; \overleftarrow{h}_i^T]^T$.

As stated in Section I, although each representation $\mathbf{h}_i$ encodes information about the i-th word with respect to all the other surrounding words in the source sentence, it lacks of an assembling mechanism to integrate these contextual information. We solve this issue with our CAEncoder in the next section.

IV. Our Approach

In this section, we first describe how to learn the future context representation (Section IV-A). Then we elaborate our context-aware recurrent encoder based on the learned context representation and input tokens (Section IV-B). Finally we show how to integrate our encoder into NMT system (Section IV-C).

A. Learning Context Representation

We mainly learn to represent the future context. For example, when reading word "*bank*" in sentence (1), the future context should include information about words "*cold water*". We choose the GRU-based RNN [6] for this task due to its capacity in memorizing processed tokens. Given a source sentence $\mathbf{x}$, the context representation $\overleftarrow{h}_i^c \in \mathbb{R}^{d_h}$ is defined as follows (We use the superscript c to denote the *context representation*):

$$\overleftarrow{h}_i^c = \text{GRU}(\overleftarrow{h}_{i+1}^c, E_{x_i}) \tag{6}$$

Here we assume that the context representation is generated from right to left. However, this is determined by the direction of CAEncoder. We explain this in Section IV-C. There are several non-linear transformations in the GRU:

$$\begin{aligned}\mathbf{z}_i &= \sigma(W_z E_{x_i} + U_z \overleftarrow{h}_{i+1}^c + b_z) \\ \mathbf{r}_i &= \sigma(W_r E_{x_i} + U_r \overleftarrow{h}_{i+1}^c + b_r) \\ \overline{h}_i^c &= \tanh(W E_{x_i} + U\left[\mathbf{r}_i \odot \overleftarrow{h}_{i+1}^c\right] + b) \\ \overleftarrow{h}_i^c &= (1 - \mathbf{z}_i) \odot \overleftarrow{h}_{i+1}^c + \mathbf{z}_i \odot \overline{h}_i^c\end{aligned} \tag{7}$$

where $\overline{h}_i^c$ is a candidate activation for the i-th token. Intuitively, GRU leverages the *reset* gate $\mathbf{r}$ and the *update* gate $\mathbf{z}$ to control whether the information should come from the past inputs or current token. All these make GRU an ideal recurrent unit for translation tasks.

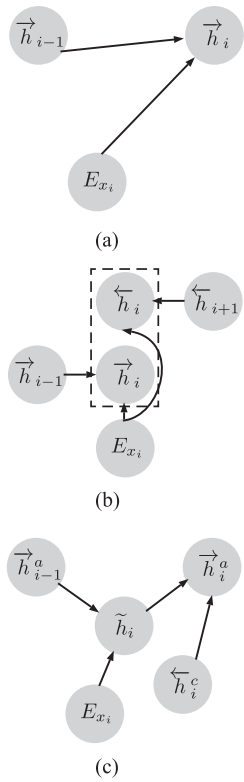

Fig. 1. Architectures of different RNN encoders. (a) Vanilla RNN. (b) Bidirectional RNN. (c) Context-aware RNN.

B. Context-Aware Recurrent Encoder

Our CAEncoder aims at combining a source sentence together with its corresponding context representations to produce a better semantic source representation. CAEncoder is essentially a variant of RNN. Considering the difference between source tokens and the future context, we design a special recurrent unit that is composed of a two-level hierarchy which is inspired by the conditional GRU decoder in *dl4mt*[2].

We illustrate our model in Fig. 1(c). Different from the vanilla recurrent model (Fig. 1(a)), our model involves a bottom level and an upper level.

The bottom level reads the source sentence to summarize the history information. Given the previous hidden state $\overrightarrow{h}_{i-1}^a$ and the current input word x_i, the bottom level generates an internal hidden state $\widetilde{h}_i$ through a GRU function (We use the superscript a to denote the *representation of CAEncoder*):

$$\widetilde{h}_i = \text{GRU}_{lower}(\overrightarrow{h}_{i-1}^a, E_{x_i}) \tag{8}$$

Intuitively, $\widetilde{h}_i \in \mathbb{R}^{d_h}$ tells the model what has been read so far until the current input word x_i.

The upper level aims at assembling the history information $\widetilde{h}_i$ provided by the bottom level and the future context $\overleftarrow{h}_i^c$ together. This is achieved by another GRU function:

$$\overrightarrow{h}_i^a = \text{GRU}_{higher}(\widetilde{h}_i, \overleftarrow{h}_i^c) \tag{9}$$

[2]https://github.com/nyu-dl/dl4mt-tutorial/tree/master/session3

Clearly, CAEncoder is an extension of the GRU-based RNN model. Unlike the bidirectional RNN model (Fig. 1(b)), CAEncoder not only preserves the capacity in handling long-term dependencies (the bottom level) but also enables the integration of future context (the upper level). Therefore, using CAEncoder for NMT can reduce the heavy burden of the decoder from assembling the interdependent forward and backward hidden states.

C. Integrating CAEncoder Into NMT

In Sections IV-A and IV-B, we assume that the history information is encoded from left to right while the encoded future context in the opposite direction. This does not matter because the history and future context are related to each other. What we have to ensure in our model is that the history and future context should be encoded in opposite directions. Accordingly, we propose two variants of the CAEncoder for NMT system:

1) *FCAEncoder* (Forward CAEncoder) CAEncoder reads a source sentence from left to right, and the future context GRU reads in an inverse direction. Formally,

$$\begin{aligned}\overleftarrow{h}_i^c &= \text{GRU}\left(\overleftarrow{h}_{i+1}^c, E_{x_i}\right)\\ \overrightarrow{h}_i^a &= \text{CAEncoder}\left(\overrightarrow{h}_{i-1}^a, E_{x_i}, \overleftarrow{h}_i^c\right)\end{aligned} \tag{10}$$

2) *BCAEncoder* (Backward CAEncoder) Both CAEncoder and the future context GRU operates in the inverse order compared with FCAEncoder. Formally,

$$\begin{aligned}\overrightarrow{h}_i^c &= \text{GRU}\left(\overrightarrow{h}_{i-1}^c, E_{x_i}\right)\\ \overleftarrow{h}_i^a &= \text{CAEncoder}\left(\overleftarrow{h}_{i+1}^a, E_{x_i}, \overrightarrow{h}_i^c\right)\end{aligned} \tag{11}$$

We use the hidden states of CAEncoder as the source representations for NMT system, i.e., $\mathbf{H} = \{\overrightarrow{h}_1^a, \ldots, \overrightarrow{h}_n^a\}/\{\overleftarrow{h}_1^a, \ldots, \overleftarrow{h}_n^a\}$. All other components, such as the attention and the decoder, are the same as the original NMT system. Notice that $\overrightarrow{h}_i^a / \overleftarrow{h}_i^a$ in CAEncoder is d_h-dimensional vector, while $\mathbf{h}_i$ in the bidirectional RNN encoder is $2d_h$-dimensional. Thus, using our CAEncoder makes the decoding more efficient.

Our CAEncoder is easy to implement, and quite efficient in training and decoding. To train our NMT system, we employ the maximum likelihood estimation as our training objective, and adopt the common stochastic gradient descent algorithm for optimization.

V. Experiments

To evaluate our approach, we conducted a series of experiments on Chinese-English and English-German translation tasks.

A. Setup

For Chinese-English translation task, our training data consists of 1.25 M sentence pairs, with 27.9 M Chinese words and 34.5 M English words respectively[3]. We used the NIST 2005 dataset as the development set, and the NIST 2002, 2003, 2004, 2006 and 2008 datasets as test sets. Translation quality is evaluated by the case-insensitive BLEU-4 metric [17][4] and TER metric [18][5]. For English-German translation task, we used the same training data of WMT 2014 which consist of 4.5 M sentence pairs with 116 M English words and 110 M German words. We used the newstest2013 (3000 sentences) as the development set, and the newstest2014 (2737 sentences) as the test set. Different from the Chinese-English translation task, we used the case-sensitive BLEU-4 metric [17] to evaluate translation quality. We performed paired bootstrap sampling [19] for significance test using the script in *Moses*[6].

We compared our proposed model against the following two widely-used systems:

1) *Moses* [20]: an open source state-of-the-art phrase-based SMT system.
2) *RNNSearch* [3]: a widely-used attention-based NMT system using bidirectional RNN as its encoder. Following the publicly available *dl4mt*[7], we implemented the decoder with two GRU layers that directly exploit the information of y_{j-1} when reading from the representation of source sentence.

For *Moses*, we used all the 1.25 M sentence pairs (without length limitation). We trained a 4-gram language model on the target portion of training data using the SRILM[8] toolkit with modified Kneser-Ney smoothing. We word-aligned the training corpus using GIZA++[9] toolkit with the option "*grow-diag-final-and*". We employed the default lexical reordering model with the type "*wbe-msd-bidirectional-fe-allff*". All other parameters were kept as the default settings.

For *RNNSearch*, we excluded the sentences that are longer than 50 (Chinese-English) and 80 (English-German) words in training. We applied the byte pair encoding compression algorithm [21] to reduce the vocabulary size as well as to deal with rich morphology. We set the vocabulary size of both source and target languages to be 30 K for Chinese-English (covering 97.7%/99.3% of the source/target side of the training data) and 40 K for English-German (covering all sentences in both the source and target side of the training data). The words that do not appear in the vocabulary were replaced with a token "*UNK*". Following Bahdanau *et al.* [3], we set $d_w = 620$, $d_h = 1000$. We initialized all parameters randomly according to a normal distribution ($\mu = 0, \sigma = 0.01$) except the square matrices which are initialized with SVD decomposition. We used the Adadelta algorithm for optimization, with a batch size of 80 and gradient norm as 5. The model parameters were selected according to the maximum BLEU points on the development set. Additionally,

[3]This data is a combination of LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

[4]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

[5]http://www.cs.umd.edu/ snover/tercom/

[6]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/analysis/bootstrap-hypothesis-difference-significance.pl

[7]https://github.com/nyu-dl/dl4mt-tutorial/tree/master/session3

[8]http://www.speech.sri.com/projects/srilm/download.html

[9]http://www.fjoch.com/GIZA++.html

TABLE I
BLEU SCORES ON THE CHINESE-ENGLISH TRANSLATION TASKS

Metric	System	MT05	MT02	MT03	MT04	MT06	MT08	AVG
BLEU	*Moses*	31.70	33.61	32.63	34.36	31.00	23.96	31.11
	RNNSearch	34.72	37.95	35.23	37.32	33.56	26.12	34.04
	Stack-RNNSearch	32.88	36.68	34.72	36.83	32.55	24.71	33.10
	FCAEncoder	36.12	39.40 $\Uparrow+$	36.99 $\Uparrow++$	39.64 $\Uparrow++$	35.32 $\Uparrow++$	**27.39**$\Uparrow+$	35.75
	BCAEncoder	36.44	**40.12**$\Uparrow++$	**37.63**$\Uparrow++$	**39.83**$\Uparrow++$	**35.44**$\Uparrow++$	27.34 $\Uparrow+$	**36.07**
TER	*Moses*	58.24	57.81	57.22	56.11	56.90	60.33	57.67
	RNNSearch	59.58	56.48	58.47	56.55	57.72	61.68	58.18
	Stack-RNNSearch	60.83	57.47	59.84	57.62	59.21	63.67	59.56
	FCAEncoder	57.01	54.39	56.74	54.02	55.73	60.08	56.19
	BCAEncoder	56.42	**53.75**	**55.82**	**53.72**	**55.66**	**60.06**	**55.80**

AVG = average BLEU score on all test sets. We highlight the best results in bold for each test set. "$\uparrow/\Uparrow$": significantly better than *Moses* ($p < 0.05/p < 0.01$); "$+/++$": significantly better than *GroundHog* ($p < 0.05/p < 0.01$). Higher BLEU scores and lower TER scores indicate better translation quality.

TABLE II
ENSEMBLE OF DIFFERENT NMT SYSTEMS

Metric	System	MT05	MT02	MT03	MT04	MT06	MT08	AVG
BLEU	*RNNSearch+FCAEncoder*	37.93	41.10	38.66	41.41	37.17	28.93	37.45
	RNNSearch+BCAEncoder	38.41	41.66	39.78	41.38	37.34	29.18	37.87
	FCAEncoder+BCAEncoder	38.67	42.14	39.86	42.03	37.33	29.74	38.22
	RNNSearch+FCAEncoder+BCAEncoder	**39.26**	**42.21**	**40.32**	**42.61**	**38.22**	**30.23**	**38.72**
TER	*RNNSearch+FCAEncoder*	55.24	52.81	54.46	53.33	54.97	59.06	54.93
	RNNSearch+BCAEncoder	55.63	53.48	55.40	53.13	54.72	59.15	55.18
	FCAEncoder+BCAEncoder	54.57	**52.15**	**53.81**	**52.34**	54.62	58.19	54.22
	RNNSearch+FCAEncoder+BCAEncoder	54.52	52.28	53.86	**52.35**	**54.06**	**58.10**	**54.13**

during decoding, we used the beam-search algorithm, and set the beam size to 10.

In addition to the vanilla *RNNSearch*, we also experimented with a *Stack-RNNSearch* which uses stacked RNN encoders for encoding. We employed two-layer RNNs, with the output of the bottom RNN being acted as the input of the upper RNN. All other settings are the same as *RNNSearch*.

For *CAEncoder*, we trained this model from scratch, i.e., randomly initialized its parameters as in *RNNSearch*. All the other settings are the same as *RNNSearch*. All NMT systems were trained on a GeForce GTX 1080 using the computational framework *Theano*.

B. Results on Chinese-English Translation

Table I shows the translation results of different systems in terms of both BLEU and TER score. RNNSearch significantly outperforms Moses by around 3 BLEU points on average. Under the TER metric, however, it performs slightly worse than Moses (0.51 TER points higher), indicating that more post-editing efforts are required for RNNSearch to exactly match reference translations. Enhanced with the proposed CAEncoder, the NMT system consistently outperforms both Moses and RNNSearch, no matter with which evaluation metric. Concretely, the FCAEncoder achieves an average gain of 1.71 BLEU points over RNNSearch and 1.48 TER points over Moses, while the BCAEncoder 2.03 BLEU points and 1.87 TER points over RNNSearch and Moses respectively. These indicate that CAEncoder is better than the bidirectional RNN encoder.[10]

Although several state-of-the-art results are reported with deep recurrent encoders [8], [10], we find that the Stack-RNNSearch yields worse results than the vanilla RNNSearch. We contribute this to the unsuitable training methods since previous works usually employ more complicated training algorithms, e.g., a combination of Adadelta and SGD. Our results further demonstrate that training such deep models are nontrivial.

It's interesting to observe that although the only difference between BCAEncoder and FCAEncoder lies on the reading direction, BCAEncoder yields marginally better translation performance. This resonates with the finding of Sutskever *et al.* [1] in that reversing the order or words in the source sentence shortens the path from the input to the output.

We also conducted additional experiments to testify whether ensembles of these NMT models can yield further improvements following previous work [22], [23]. The results are summarized in Table II. All these ensemble models achieve very promising improvements, a gain of around 1.5 BLEU points over the single best system. We find that the ensemble of BCAEncoder and FCAEncoder outperforms the ensemble of CAEncoder and RNNSearch. This may be because BCAEncoder and

[10]We also tried to combine the forward and backward representations in bidirectional RNN using GRU function, rather than the simple concatenation. However, results show almost no gains against the original RNNSearch.

TABLE III
STATISTICS ABOUT THE NMT MODELS ON CHINESE-ENGLISH TRANSLATION TASKS

Model	#Params	Train Speed	Test Speed
RNNSearch	89.67 M	0.91 s	1.07 s
FCAEncoder	87.05 M	0.82 s	0.98 s
BCAEncoder	87.05 M	0.83 s	0.95 s

#Params = the number of model parameters. **Train Speed** = the average time in seconds for training one batch; **Test Speed** = the average time in seconds for decoding one source sentence.

FCAEncoder are both based on CAEncoder but operate in opposite directions. Therefore, the complementarity between them is much stronger. Together with RNNSearch, the all-in ensemble system obtains further improvements, and reaches 38.27 BLEU points and 54.13 TER points on average.

C. Model Analysis

In this section, we provide more details of our models in terms of the model parameters and computation cost. The results are summarized in Table III. Except for better performance, both our models have fewer model parameters and faster training and decoding speed than the baseline. This strongly shows that our model is as efficient as the vanilla bidirectional RNN encoder.

D. Effects Over Sentence Lengths

Following Bahdanau *et al.* [3], we group sentences of similar lengths together to evaluate the NMT models. We divide our test sets into 6 disjoint groups according to the length of source sentences ((0, 10), [10, 20), [20, 30), [30, 40), [40, 50), [50, -)), each of which contains 680, 1923, 1839, 1189, 597 and 378 sentences respectively. Fig. 2 illustrates the overall results. We find that NMT models perform better on short sentences than long sentences. On the longest sentences (their length is longer than 50, the upper bound length of training instances), the performance of NMT models degrades significantly with a drop of around 7 BLEU points. This finding is consistent with that of previous work [3], [4]. Our CAEncoder is also confronted with this long sentence translation problem, but behaves more robust. Specifically, our model outperforms RNNSearch across all sentence length groups.

Additionally, we also provide the average length of translations for each model on different groups (Fig. 2 right part). We find that the average length of different NMT models is almost the same. This strongly indicates that neither longer nor shorter translations lead to the improvements of our CAEncoder. We argue that our model improves translation quality by making translations more faithful with respect to source sentences. We verify this point in the next subsection.

Different from the NMT models, the performance of Moses does not significantly drop as the length of source sentences increases. Specifically, Moses obtains the best translation performance on the longest sentence group. We argue that this is because of the phrase-based translation philosophy of Moses which enables it to yield faithful translations even for long source sentences. Notice that on the longest sentence group, the average length of Moses-generated translations is about 67 words, while that of NMT-generated translations is only about 55.

E. Translation Analysis

Why does our CAEncoder obtain such significant improvements? To answer this question, we dig into translations and provide some translation examples in Table IV.

The translations of Moses tend to be faithful but not fluent. For example, *for the continent* in the second example is incorrectly placed at the front of the sentence in translation "*. . . new century for the continent has . . .*". In contrast, the translations of NMT systems are more fluent. However, they sometimes incorrectly convey the meanings of the source sentences, generating unfaithful translations. For example, in the first translation, RNNSearch translates in a way that the two girls "*were not parents*" rather than "*are without parents*". In the second translation, RNNSearch fails to recognize that the "*better tomorrow*" is "*for africa*". In the third translation, RNNSearch thinks that "*the increasingly frequent bombings*" happen in "*palestinian areas*", rather than made by "*palestinian*". These translations change the meanings of source sentences, which is rather undesirable.

Our CAEncoder, although its generated translations are not perfect either, handles these problems much better. In the first translation, it learns that the "*two girls do not have parents*". While, in the second translation, it successfully recognizes that the contributions should lead to "*the better future in africa*". And in the third translation, it believes that the "*the increasingly frequent bombing activities*" are created by "*palestine*". We find that the length of the translation generated by different NMT systems does not differ so much, but the semantics of different translations differs very significantly. This suggests that our model achieves better translation performance because it is more capable of dealing with the underlying semantics of source sentences. And we contribute this to the way that CAEncoder assembles the history and future context representations.

We further show the attention weights (see (4)) in different models to verify if this unfaithfulness is caused by a wrong alignment. Due to the limit of space, we only visualize the first example in Table IV. The results are shown in Fig. 3. We find that the attention weights generated by different models are very similar. This is reasonable because we use the same attention mechanism in CAEncoder and RNNSearch. Dissimilarly, RNNSearch aligns the source word "没有" to "*were not*", while our CAEncoder aligns it to "*do not have*". Although both translations involve the key word "*not*", the overall meaning is completely different. Excluding the differences in the attention and decoder, we conclude that our CAEncoder is better at modeling the semantics of source sentences.

F. Results on English-German Translation

We provide the results of our systems on English-German translation task as well as several previous systems (including the winning system in WMT14 [24], a phrase-based system whose language models were trained on a huge monolingual

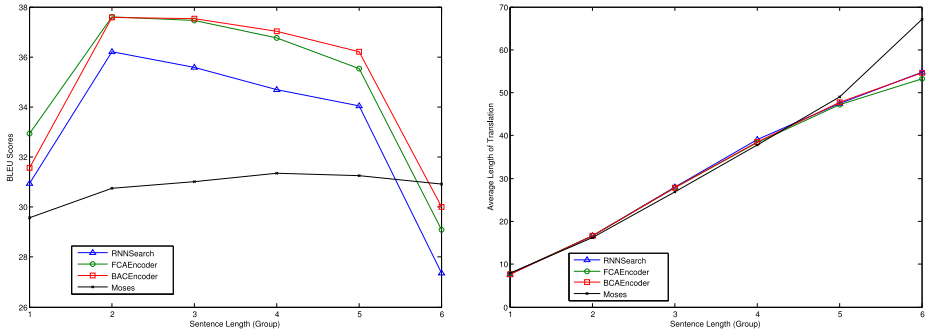

Fig. 2. Translation results on the test sets with respect to the lengths of the source sentences. The left figure shows the BLEU score, and the right one shows the average length of translations.

TABLE IV
SOME EXAMPLES GENERATED BY DIFFERENT SYSTEMS

Source	他们 来自 六 个 家庭 , 其中 两 个 女孩 没有 父母 。
Reference	*they came from six families and two girls are without parents.*
Moses	*they came from six families, including two girls not parents.*
RNNSearch	*they came from six families, of which two girls were not parents.*
CAEncoder	*they come from six families, of which two girls do not have parents.*
Source	新 世纪 的 到来 为 非洲 大陆 带来 了 诸多 机遇 与 挑战 , 非洲 青年人 应 努力 掌握 高 新 技术 , 为 非洲 更 美好 的 明天 做出 自己 的 贡献 。
Reference	*the advent of the new century has brought many opportunities and challenges to the african continent. the african youth should work hard to master the high and new technologies in order to make their due contribution to the more beautiful tomorrow of africa.*
Moses	*the arrival of the new century for the continent has brought many opportunities and challenges, the african young people should make efforts to master the new and high technology, and to africa more beautiful tomorrow.*
RNNSearch	*the arrival of the new century has brought many opportunities and challenges for the african continent, and african youths should make efforts to seize new high technologies and make their own contributions to the better tomorrow.*
CAEncoder	*the arrival of the new century has brought many opportunities and challenges to the african continent, and the young people of africa should strive to master high technology and make their own contributions to the better future in africa.*
Source	总理 沙龙 当天 下午 召集 核心 部长 就 以 巴 当前 局势 , 以及 如何 应对 巴勒斯坦 方面 日益 频繁 的 爆炸 袭击 活动 进行 磋商 , 会议 决定 加大 对 巴方 的 军事 打击 行动 。
Reference	*prime minister ariel sharon called a cabinet meeting in the afternoon of 31st to discuss the situation between israel and palestinian. the cabinet discussed on how to deal with the increasing bomb attacks by palestinian. the meeting decided to increase military actions against palestinian.*
Moses	*israeli prime minister sharon on the afternoon called on israel and palestine, the core of the current situation, and how to cope with the palestinian side bombing attacks increasingly frequent consultations, the meeting decided to increase the military action against the palestinians.*
RNNSearch	*on the afternoon of the same day, prime minister sharon called the core minister on the current situation in palestine, and on how to cope with the increasingly frequent bombings in palestinian areas, the meeting decided to increase military attacks against the palestinians.*
CAEncoder	*on the afternoon of the same day, israeli prime minister sharon called the core ministers to hold consultations on the current situation in palestine and how to deal with the increasingly frequent bombing activities of palestine, and decided to increase military attacks against palestine.*

CAEncoder is more faithful to the meanings of source sentences. Important phrases are highlighted in red color.

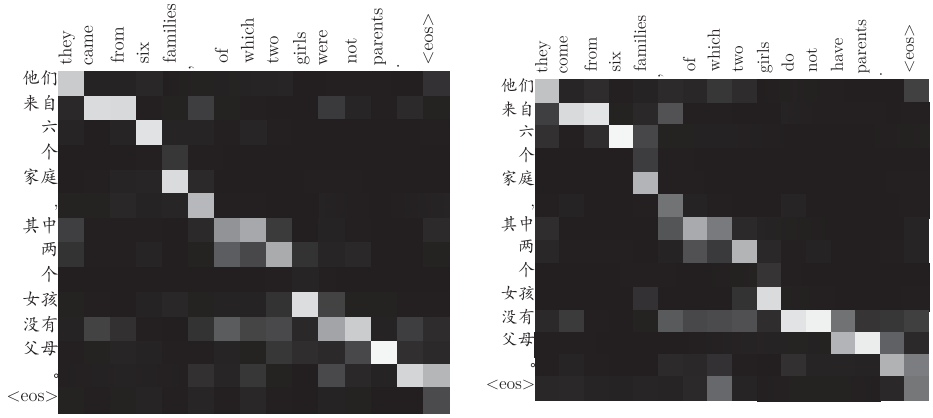

Fig. 3. Visualization of the attention weights for RNNSearch (left) and CAEncoder (right).

TABLE V
CASE-SENSITIVE BLEU SCORES ON THE ENGLISH-GERMAN TRANSLATION TASK

System	Architecture	Vocab	BLEU
Buck *et al.* [24]	Winning WMT14 system phrase-based + large LM	-	20.7
	Existing end-to-end NMT systems		
Jean *et al.* [22]	RNNSearch + unk replace + large vocab	500 K	19.40
Luong *et al.* [23]	LSTM with 4 layers + dropout + local att. + unk replace	50 K	20.90
Shen *et al.* [4]	RNNSearch (GroundHog) + MRT + PosUnk	50 K	20.45
Zhou *et al.* [8]	LSTM with 16 layers + Fast-Forward connections	80 K	20.60
Wu *et al.* [10]	LSTM with 8 layers + RL-refined Word	80 K	23.10
Wu *et al.* [10]	LSTM with 8 layers + RL-refined WPM	16 K	24.36
Wu *et al.* [10]	LSTM with 8 layers + RL-refined WPM	32 K	24.61
Wang *et al.* [25]	RNNSearch with 4 layers + LAU	80 K	22.10
Wang *et al.* [25]	RNNSearch with 4 layers + LAU + PosUnk	80 K	23.80
	Our end-to-end NMT systems		
this work	RNNSearch + BPE	40 K	20.87
	FCAEncoder + BPE	40 K	21.86
	BCAEncoder + BPE	40 K	22.57

"*unk replace*" and "*PosUnk*" denote the approaches of handling rare words in Jean *et al.* [22] and Luong *et al.* [23] respectively. "*RL*" and "*WPM*" represent the reinforcement learning optimization and wordpiece model used in Wu *et al.* [10], respectively. "*LAU*" and "*MRT*" denote the linear associative unit and the minimum risk training proposed by Wang *et al.* [25] and Shen *et al.* [4] respectively. "*BPE*" denotes the byte pair encoding algorithm in Sennrich *et al.* [21].

text, the Common Crawl corpus) in Table V. Our RNNSearch model, without any specific features, yields a BLEU score of 20.87, a very competitive result compared against several advanced systems [4], [8], [22], [23]. Enhanced with the proposed CAEncoder, both our models outperform the RNNSearch with very significant improvements. Specially, the BCAEncoder achieves a BLEU score of 22.57, making it comparable to some strong deep models [10], [25]. These encouraging results demonstrate that our model can be applied to different language pairs.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented a context-aware recurrent encoder (CAENcoder) for NMT systems. As an alternative to the conventional bidirectional recurrent encoder, our CAEncoder learn to assemble the future and history context information in a better way to model the semantics of source sentences such that the yielded source representations are more expressive. To achieve this, CAEncoder first employs a GRU-based RNN to model future context representations in advance, and then uses a two-level GRU model to process the future context representations and input tokens jointly. We further propose two variants, namely FCAEncoder and BCAEncoder according to the encoding direction. Experiments on both Chinese-English and English-German translation tasks demonstrate the effectiveness of our model.

In the future, we would like to extend CAEncoder in three directions: 1) Applying our CAEncoder to other sequence-to-sequence tasks, e.g., dialog, question answering and document summarization and examing its effectiveness on these tasks. 2) Attempting to deepen our CAEncoder but still preserve its efficiency in training and optimization. 3) Extending CAEncoder from sentence-level translation to document-level translation with wider context information.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 27th Int. Conf. Neural Informat. Proces. Syst.*, 2014, vol. 2, pp. 3104–3112.
[2] K. Cho *et al.*, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Doha, Qatar, Oct. 2014, pp. 1724–1734.
[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2014.
[4] S. Shen *et al.*, "Minimum risk training for neural machine translation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, pp. 1683–1692, Aug. 2016.
[5] B. Zhang, D. Xiong, J. Su, H. Duan, and M. Zhang, "Variational neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, Nov. 2016, pp. 521–530. [Online]. Available: https://aclweb.org/anthology/D16–1050
[6] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," NIPS Workshop on Deep Learning, Dec. 2014.
[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.
[8] J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu, "Deep recurrent models with fast-forward connections for neural machine translation," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 371–383, 2016.
[9] F. Meng, Z. Lu, Z. Tu, H. Li, and Q. Liu, "Neural transformation machine: A new architecture for sequence-to-sequence learning," arXiv:1506.06442, 2015.
[10] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," arXiv:1609.08144, 2016.
[11] A. Eriguchi, K. Hashimoto, and Y. Tsuruoka, "Tree-to-sequence attentional neural machine translation," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, Aug. 2016, pp. 823–833.
[12] J. Su, Z. Tan, D. Xiong, and Y. Liu, "Lattice-based recurrent neural network encoders for neural machine translation," 2017, pp. 3302–3308.
[13] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," arXiv:1610.10099, 2016.

[14] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A convolutional encoder model for neural machine translation," *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 123–135, Jul. 2017.
[15] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[16] H. Choi, K. Cho, and Y. Bengio, "Context-dependent word representation for neural machine translation," *Comput. Speech & Lang.*, vol. 45, pp. 149–160, 2017.
[17] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.
[18] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proc. Assoc. Mach. Transl. Amer.*, 2006, pp. 223–231.
[19] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2004, pp. 388–395.
[20] P. Koehn *et al.*, "Moses: Open source toolkit for statistical machine translation," in *Proc. 45th Annu. Meeting Assoc. Comput. Linguistics*, 2007, pp. 177–180.
[21] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Aug. 2016, pp. 1715–1725.
[22] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On using very large target vocabulary for neural machine translation," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process. Asian Fed. Natural Lang. Process.*, Jul. 2015, pp. 1–10.
[23] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Sep. 2015, pp. 1412–1421.
[24] C. Buck, K. Heafield, and B. van Ooyen, "N-gram counts and language models from the common crawl," in *Proc. Lang. Resour. Eval. Conf.*, Reykjavik, Iceland, May 2014, pp. 3579–3584.
[25] M. Wang, Z. Lu, J. Zhou, and Q. Liu, "Deep Neural Machine Translation with Linear Associative Unit," *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, pp. 136–145, Jul. 2017.

Biao Zhang received the Bachelor's degree in software engineering from Xiamen University, Xiamen, China. He is currently a graduate student in the School of Software, Xiamen University. He is supervised by Prof. H. Duan and Prof. J. Su. His major research interests include natural language processing and deep learning.

Deyi Xiong received the Ph.D. degree in computer science from the Institute of Computing Technology, of the Chinese Academy of Sciences, Beijing, China, in 2007. He is currently a Professor at Soochow University, Suzhou, China. Previously, he was a Research Scientist in the Institute for Infocomm Research of Singapore from 2007 to 2013. His research interests include the area of natural language processing, including parsing, and neural machine translation.

Jinsong Su was born in 1982. He received the Ph.D. degree from the Chinese Academy of Sciences, Beijing, China. He is currently an Associate Professor in Software School, Xiamen University, Xiamen, China. His research interests include natural language processing and neural machine translation.

Hong Duan was born in 1976. He received the Ph.D. degree from the University of Science and Technology of China, Hefei, China. He is currently an Associate Professor in Software School, Xiamen University, Xiamen, China. His research interests include machine learning and virtual reality.