Context-Aware Smoothing for Neural Machine Translation

Kehai Chen[1]*, **Rui Wang**[2]†, **Masao Utiyama**[2], **Eiichiro Sumita**[2] **and Tiejun Zhao**[1]
[1]Machine Intelligence & Translation Laboratory, Harbin Institute of Technology
[2]ASTREC, National Institute of Information and Communications Technology (NICT)
{khchen and tjzhao}@hit.edu.cn
{wangrui, mutiyama and eiichiro.sumita}@nict.go.jp

Abstract

In Neural Machine Translation (**NMT**), each word is represented as a low-dimension, real-value vector for encoding its syntax and semantic information. This means that even if the word is in a different sentence context, it is represented as the fixed vector to learn source representation. Moreover, a large number of Out-Of-Vocabulary (**OOV**) words, which have different syntax and semantic information, are represented as the same vector representation of *unk*. To alleviate this problem, we propose a novel context-aware smoothing method to dynamically learn a sentence-specific vector for each word (including OOV words) depending on its local context words in a sentence. The learned context-aware representation is integrated into the NMT to improve the translation performance. Empirical results on NIST Chinese-to-English translation task show that the proposed approach achieves 1.78 BLEU improvements on average over a strong attentional NMT, and outperforms some existing systems.

1 Introduction

Neural Machine Translation (**NMT**) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015), has shown prominent performances in comparison with the conventional Phrase Based Statistical Machine Translation (**PBSMT**) (Koehn et al., 2003). In NMT, a source sentence is converted into a vector representation by an RNN called *encoder*, then another RNN called *decoder* generates target sentence word by word based on the source representation with attention information and target history.

One advantage of NMT systems is that each word is represented as a low-dimension, real-valued vector, instead of storing statistical rules as in PBSMT. This means that even if the word is in a different sentence context, it is represented as the fixed vector to learn source representation. Figure 1 (a) shows two pair of Chinese-to-English parallel sentences in which two Chinese sentences contain the same word "*da*". Intuitively, the "*da*" denotes "beating" in the first Chinese sentence while the "*da*" denotes "playing" in the second Chinese sentence. It is obvious that the "*da*" which denotes different meanings in a specific sentence is represented as the same word vector in the encoder of NMT, as show in Figure 1 (b). Although the RNN-based encoder can capture the sentence context for each word, we believe that offering better word vector with context-aware representation might help improve translation quality of NMT.

Moreover, a large number of Out-Of-Vocabulary (**OOV**) words which have different syntax and semantic information are represented as the same vector representation of *unk*. Actually, this kind of simple approach may cause ambiguity of the sentences since the single *unk* breaks the structure of sentences, thus hurts representation learning of source sentence and translation prediction of the target word. For example, the *unk* firstly affects source representation learning in *encoder*; then the negative effect would be further transformed to the *decoder*, which generates the poverty context vector and hidden layer state for translation prediction, as shown in the gray parts of Figure 1 (c). Besides, when the generated target word may also be *unk*, the negative effect of *unk* will become more severe.

*Kehai Chen was an internship research fellow at NICT when conducting this work.
†Corresponding author.

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 11–20,
Taipei, Taiwan, November 27 – December 1, 2017

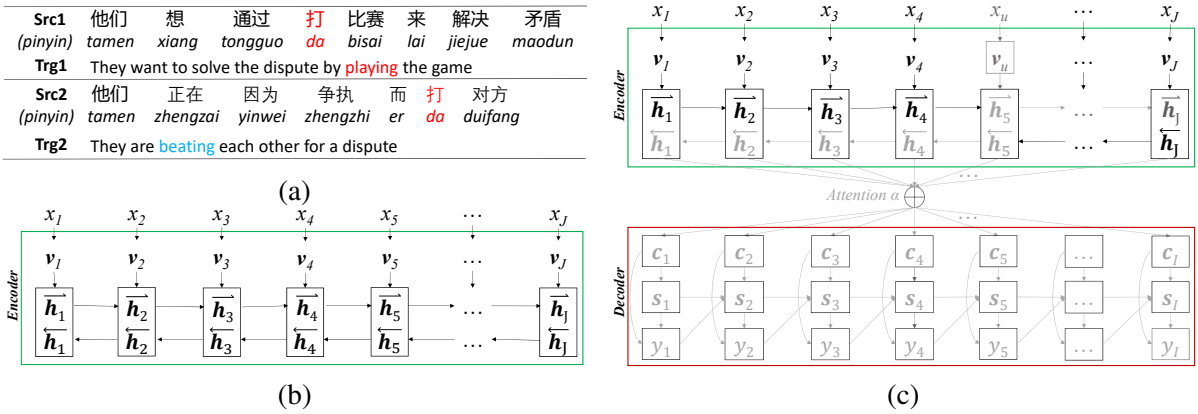

Figure 1: (a) Two parallel Chinese-to-English sentence pair; (b) The encoder of NMT; (c) The NMT with OOV, these gray parts indicate the parameters of NMT which are affected by the OOV x_u.

In this paper, we propose a novel context-aware smoothing method to dynamically learn a Context-Aware Representation (**CAR**) for each word (including OOV words) depending on its local context words in a sentence. We then use the learned CAR to extend word vector in a sentence, thus enhancing source representation for improving the translation performance of NMT. First, compared with the single *unk* vector, we encode the context words of each OOV as a Context-Aware Representation (**CAR**), which has the potential to capture the *OOV*'s semantic information. Second, we also extend the context-aware smoothing method to in-vocabulary words, which enhances *encoder* and *decoder* of NMT by more effectively utilizing context information by the learned CAR. To this end, we proposed two unique neural networks to learn the context-aware representation for each word depending on its context words in a fixed-size window. We then design four NMT models with CAR to improve translation performance by smoothing the encoder and decoder.

The remainder of the paper is organized as follows. Section 2 introduces the related work in the NMT. Section 3 presents two novel neural models to learn the CAR for each word. Section 4 integrates the CAR into the NMT by using smoothing strategies. Section 5 reports the experimental results obtained in the Chinese-to-English task. Finally, we conclude the contributions of the paper and discuss the further work in Section 6.

2 Related Work

In traditional SMT, there are many research works to improve the translations of OOVs. Fung and Cheung (2004) and Shao and Ng (2004) adopte comparable corpora and web resources to extract translations for each unknown word. Marton et al. (2009) and Mirkin et al. (2009) applied paraphrase model and entailment rules to replace unknown words with in-vocabulary synonyms before translation. A series of works (Knight and Graehl, 1997; Jiang et al., 2007; Al-Onaizan and Knight, 2002) utilized transliteration and web mining techniques with external monolingual/bilingual corpora, comparable data and the web resource to find the translation of the unknown words. Nearly most of the related PBSMT researches focused on finding the correct translation of the unknown words with external resources and ignored the negative effect for other words.

Compared with PBSMT, due to high computational cost, NMT has a more limited vocabulary size and severe OOV phenomenon. The existing PBSMT methods that used external resources to translate unknown words for SMT are hard to be directly introduced into NMT, because of NMT's *soft*-alignment mechanism (Bahdanau et al., 2015). To relieve the negative effect of unknown words for NMT, Luong et al. (2015) proposed a word alignment algorithm, allowing the NMT system to emit, for each OOV word in the target sentence, the position of its corresponding word in the source sentence, and to translate every OOV in a post-processing step using a external bilingual dictionary. Although

these methods improved the translation of OOV, they must learn external bilingual dictionary information in advance.

From the point of vocabulary size, many works tried to use a large vocabulary size, thus covering more words. Jean et al. (2015) proposed a method based on importance sampling that allowed NMT model to use a very large target vocabulary for relieving the OOV phenomenon in NMT, which are only designed to reduce the computational complexity in training, not for decoding. Arthur et al. (2016) introduced discrete translation lexicons into NMT to imrpove the translations of these low-frequency words. Mi et al. (2016) proposed a vocabulary manipulation approach by limiting the number of vocabulary being predicted by each batch or sentence, to reduce both the training and the decoding complexity. These methods focused on the translation of OOV itself and ignored the other negative effect caused by the OOV, such as the translations of the words around the OOV.

Recently, many works exploited the granularity translation unit from words to smaller subwords or characters. Sennrich et al. (2016) introduced a simpler and more effective approach to encode rare and unknown words as sequences of subword units by Byte Pair Encoding (Gage, 1994). This is based on the intuition that various word classes are translatable via smaller units than words. Luong and Manning (2016) segmented the known words into character sequence, and learned the unknown word representation by character-level recurrent neural networks, thus achieving open vocabulary NMT. Li et al. (2016) replaced OOVs with in-vocabulary words by semantic similarity to reduce the negative effect for words around the OOVs. Costa-jussà and Fonollosa (2016) presented a character-based NMT, in which character-level embeddings were in combination with convolutional and highway layers to replace the standard lookup-based word representations. These methods extended the vocabulary to a larger or unlimited vocabulary and improved the performance of NMT tasks, especially in the morphological rich language pairs.

Instead of utilizing larger vocabulary or sub-unit information, we exploit to relieve more translation performance for NMT from the negative effect of OOVs by learning context-aware representations for OOVs. As a result, the proposed method can smooth the representation of word and reduce the *unk*'s negative effect in attention model, context annotations and decoding hidden states, thus improving the performance of NMT.

3 Context-Aware Representation

Intuitively, when one understands natural language sentence, especially including polysemy words or OOVs, one often inferences the meaning of these words depending on its context words. Context plays an important role in learning distributed representation of word (Mikolov et al., 2013a,b). Motivated by this, we propose two neural network methods, including Feedforward Context-of-Word Model (**FCWM**) and Convolutional Context-of-Words Model (**CCWM**), to learn a Context-Aware Representation (**CAR**) for each word.

3.1 Feedforward Context-of-Words Model

Inspired by the representation learning of word (Bengio et al., 2003), the proposed FCWM includes an input layer, a projection layer, and a non-linear output layer, as shown in Figure 2 (a).

Specifically, suppose there is a source language sentence, $\{x_1, x_2, \ldots, x_j, \ldots, x_J\}$. If the context window is set as $2n$ ($n = 2$), the context of each word x_i is defined as its historical n words and future n words:

$$L_j = x_{j-n}, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{j+n}. \quad (1)$$

In the input layer, each word in L_j is transformed into one-hot representation. [1] The projection layer concatenates one-hot representations in L_j to a $(2nm)$-dimension vector $\boldsymbol{L}_j$,[2]

$$\boldsymbol{L}_j = [v_{j-n} : \ldots, v_{j-1} : v_{j+1} : \cdots : v_{j+n}], \quad (2)$$

where ":" denotes the concatenation operation of word vectors.

We then approximate to learn its semantic representation $\boldsymbol{V}_{L_j} \in R^m$ by a non-linear output layer instead of softmax layer:

$$\boldsymbol{V}_{L_j} = \sigma(\boldsymbol{W}_1 \boldsymbol{L}_j + \boldsymbol{b}_1)^T, \quad (3)$$

[1] If the L_j includes OOV, we use original *unk* vector to represent it. Besides, we also try the average vector of the current sentence word to represent it, but gain the similar translation performance.

[2] In this paper, the bold variable denotes a continuous space vector.

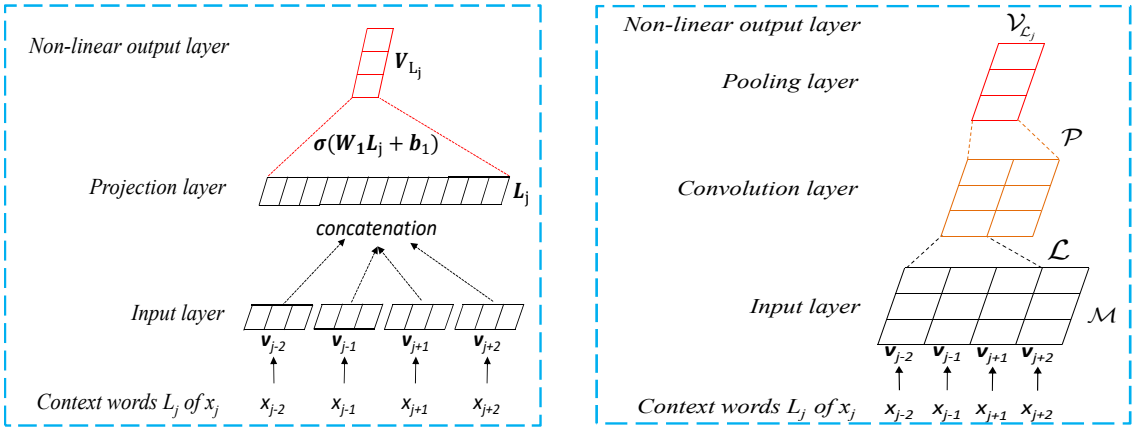

Figure 2: (a) Feedforward Context-of-Word Model; (b) Convolution Context-of-Word Model.

where σ is a non-linear activation function (e.g., *Tanh*), T represents matrix transpose, and $\boldsymbol{W}_1$ is a weight matrix and $\boldsymbol{b}_1$ is a bias term.

Finally, we extend each word with the learned CAR vector $\boldsymbol{V}_{L_j}$, thus feeding into the NMT to enhance source representation for improving target word prediction. Therefore, the proposed FCWM plays the role of the function φ parameterized by θ_1, which maps the context L_j of each word into vector $\boldsymbol{V}_{L_j}$ as follows:

$$\boldsymbol{V}_{L_j} = \varphi(L_j; \theta_1). \quad (4)$$

3.2 Convolutional Context-of-Words Model

Compared with the FCWM, the proposed CCWM indirectly encodes the context words of each word as a compositional semantic representation to represent the OOV. Specifically, the proposed CCWM is a novel variant of the standard convolutional neural network (Collobert et al., 2011), including an input layer, a convolution layer, a pooling layer and a non-linear output layer, as shown in Figure 2 (b).

Input Layer: When the dimension of word vector is m and the context window is set to $2n$, the input layer is denoted as one vector matrix $\mathcal{M} \in R^{m \times 2n}$. In $\mathcal{M}$, each column denotes context words of word x_j, that is, $\mathcal{M}$ is $[\boldsymbol{v}_{j-n}, \cdots, \boldsymbol{v}_{j-1}, \boldsymbol{v}_{j+1}, \cdots, \boldsymbol{v}_{j+n}]$ for the context $\{x_{j-n}, \cdots, x_{j-1}, x_{j+1}, \cdots, x_{j+n}\}$ of x_j.

Convolutional Layer: In the convolutional layer, let the filter window size be $m \times k$ ($2 \leq k \leq 2n$), where the k is set to 3 in our experiments, thus generating feature map $\boldsymbol{\mathcal{L}}_j$ as follows:

$$\boldsymbol{\mathcal{L}}_j = \psi(\boldsymbol{W}_2[\boldsymbol{v}_j : \boldsymbol{v}_{j+1} : \cdots : \boldsymbol{v}_{j+k}] + \boldsymbol{b}_2), \quad (5)$$

where ψ is a non-linear activation function,[3] $\boldsymbol{W}_2 \in R^{m \times k \cdot m}$ is the weight matrix and $\boldsymbol{b}_2 \in R^m$ is a bias term. After the filter traverses the input matrix, the output of the feature map $\boldsymbol{\mathcal{L}}$ is:

$$\boldsymbol{\mathcal{L}} = [\boldsymbol{\mathcal{L}}_1, \ldots, \boldsymbol{\mathcal{L}}_{2n-k+1}]. \quad (6)$$

Pooling Layer: The pooling operation (e.g., *max*, *average*) is commonly used to extract robust features from convolution. For the output feature map of the convolution layers, a column-wise *max* is performed over the consecutive columns of window size 2 as follows:

$$\boldsymbol{\mathcal{P}}_l = max[\boldsymbol{\mathcal{L}}_{2l-1}, \boldsymbol{\mathcal{L}}_{2l}], \quad (7)$$

where $1 \leq l \leq \frac{2n-k+1}{2}$. After the *max* pooling, the output of the feature map $\mathcal{P}$ is:

$$\boldsymbol{\mathcal{P}} = [\boldsymbol{\mathcal{P}}_1, \ldots, \boldsymbol{\mathcal{P}}_{\frac{2n-k+1}{2}}]. \quad (8)$$

Non-linear Output Layer: The output layer is typically a fully connected layer multiplied by a matrix. In this paper, first row-wise averaging from the pooling layers is performed without any parameters, and gain CAR of each word by non-linear active function σ (e.g., Tanh); hence, the CAR $\boldsymbol{\mathcal{V}}_{\boldsymbol{\mathcal{L}}_j}$ of word x_j is obtained by

$$\boldsymbol{\mathcal{V}}_{\boldsymbol{\mathcal{L}}_j} = \sigma(\boldsymbol{W}_3(average(\sum_{l=1}^{\frac{2n-k+1}{2}} \boldsymbol{\mathcal{P}}_l)) + \boldsymbol{b}_3). \quad (9)$$

Therefore, the above CCWM plays the role of the function φ parameterized by θ_2, which maps

[3] We used a ReLU activation function.

the context $\mathcal{L}_j$ of word x_j into vector $\boldsymbol{\mathcal{V}}_{\mathcal{L}_j}$ as follows:

$$\boldsymbol{\mathcal{V}}_{\mathcal{L}_j} = \varphi(\boldsymbol{\mathcal{L}}_j; \theta_2) \tag{10}$$

In this case, the word x_j is represented as a CAR $\boldsymbol{\mathcal{V}}_{\mathcal{L}_j}$.

4 NMT with Context-Aware Smoothing

4.1 NMT Background

An NMT model consists of an *encoder* process and a *decoder* process, and hence it is often called *encoder-decoder* model (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015), as shown in Figure 1. Typically, each unit of source input $(x_1, \ldots, x_J)$ is firstly embedded as a vector $\boldsymbol{v}_{x_j}$, and then represented as annotation vector $\boldsymbol{h}_j$ by

$$\boldsymbol{h}_j = f_{enc}(\boldsymbol{v}_{x_j}, \boldsymbol{h}_{j-1}), \tag{11}$$

where f_{enc} is a bidirectional Recurrent Neural Network (**RNN**) (Bahdanau et al., 2015). These annotation vectors $\{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_J\}$ are used to generate target word in *decoder*.

An RNN *decoder* is used to compute the target word y_i probability by a softmax layer g:

$$P(y_i|y_{<i}, x) = g(\boldsymbol{v}_{y_{i-1}}, \boldsymbol{s}_i, \boldsymbol{c}_i), \tag{12}$$

where $\boldsymbol{v}_{y_{i-1}}$ is vector representation of the previously emitted word y_{i-1}, $\boldsymbol{s}_i$ is an RNN hidden state for the current time step and the $\boldsymbol{c}_i$ is the current context vector.

4.2 Smoothing Strategy

In this subsection, we will introduce NMT with the learned CAR. This would relieve the translation performance of NMT from source representation. To this end, we use OOV as an example to integrate FCWM or CCWM into NMT; and then extend them to in-vocabulary words.

To learn the representation of source sentence, the proposed FCWM or CCWM are integrated into the *encoder* of NMT. If the source word x_j is in-vocabulary, its annotation vector $\boldsymbol{h}_j$ is learned by the traditional *encoder*; if the source word x_j is not in-vocabulary (OOV x_u), the FCWM or CCWM proposed in section 3 are used to learn its CAR instead of single *unk* vector, and further learn its annotation vector $\boldsymbol{h}_j$. According to the eq.(11), the *encoder* with CAR learns the annotation vector $\boldsymbol{h}_j$ by the eq.(13):

$$\boldsymbol{h}_j = \begin{cases} f_{enc}(\boldsymbol{h}_{x_j}, \boldsymbol{h}_{j-1}), & x_j \in V_s \\ f_{enc}(\varphi_e(\boldsymbol{V}_{L_{x_j}}), \boldsymbol{h}_{j-1}), & x_j \notin V_s, \end{cases} \tag{13}$$

where V_s is source-side vocabulary table in NMT, φ_e is the proposed FCWM or CCWM integrated into the *encoder* according to eq.(4) or eq.(10), and $\boldsymbol{V}_{L_j}$ is the learned CAR over the source-side L_j from eq.(1):

$$L_j = x_{j-n}, \ldots, x_{j-1}, x_{j+1}, \ldots, x_{j+n}.^{4} \tag{14}$$

Similarly, the proposed FCWM or CCWM are also integrated into the *decoder* in NMT. Compared with the *encoder* with CAR, the target-side OOV's context words of training processing is different from that of the decoding in which target-side OOV's future context is unknowable. That is, only the historical n words of y_{i-1} are used to learn the CAR of $\boldsymbol{V}_{L'_{i-1}}$. To be consistent with the decoding process, the previous $2n$ words of OOV are regarded as its context L'_{i-1} instead of the previous n words and future n words. Therefore, the *decoder* with CAR predicts the next target word by the eq.(15):

$$P(y_i|y_{<i}, x) = \begin{cases} g(\boldsymbol{v}_{y_{i-1}}, \boldsymbol{s}_i, \boldsymbol{c}_i), & y_{i-1} \in V_t \\ g(\varphi_d(L'_{i-1}), \boldsymbol{s}_i, \boldsymbol{c}_i), & y_{i-1} \notin V_t, \end{cases} \tag{15}$$

where V_t is target-side vocabulary table in NMT, φ_d denotes the proposed FCWM or CCWM integrated into the *decoder* according to eq.(4) or eq.(10), and $\boldsymbol{V}_{L'_{i-1}}$ is the learned CAR over the target-side context L'_{i-1} from eq.(1):

$$L'_{i-1} = y_{i-2n}, \ldots, y_{i-n}, \ldots, y_{i-1}.^{5} \tag{16}$$

4.3 Models

Based on the above smoothing strategy, we design four novel NMT models: **CARNMT**-Encoder, **CARNMT**-Decoder, **CARNMT**-Both and an **ALLSmooth**, all of which can make use of CAR to enhance *encoder* or *decoder* of NMT for improving the translation performance:

- **CARNMT**-Encoder: Only smoothing source-side unk to relieve the influence in the *encoder*, as shown in Figure 3 (a).

[4] If the number of previous context or future context words is less n, we pads a sentence start symbol *BEG* or sentence end symbol *EOS*.

[5] If the number of previous context words is less $2n$, we pads L_{i-1} using a sentence start symbol *BEG*.

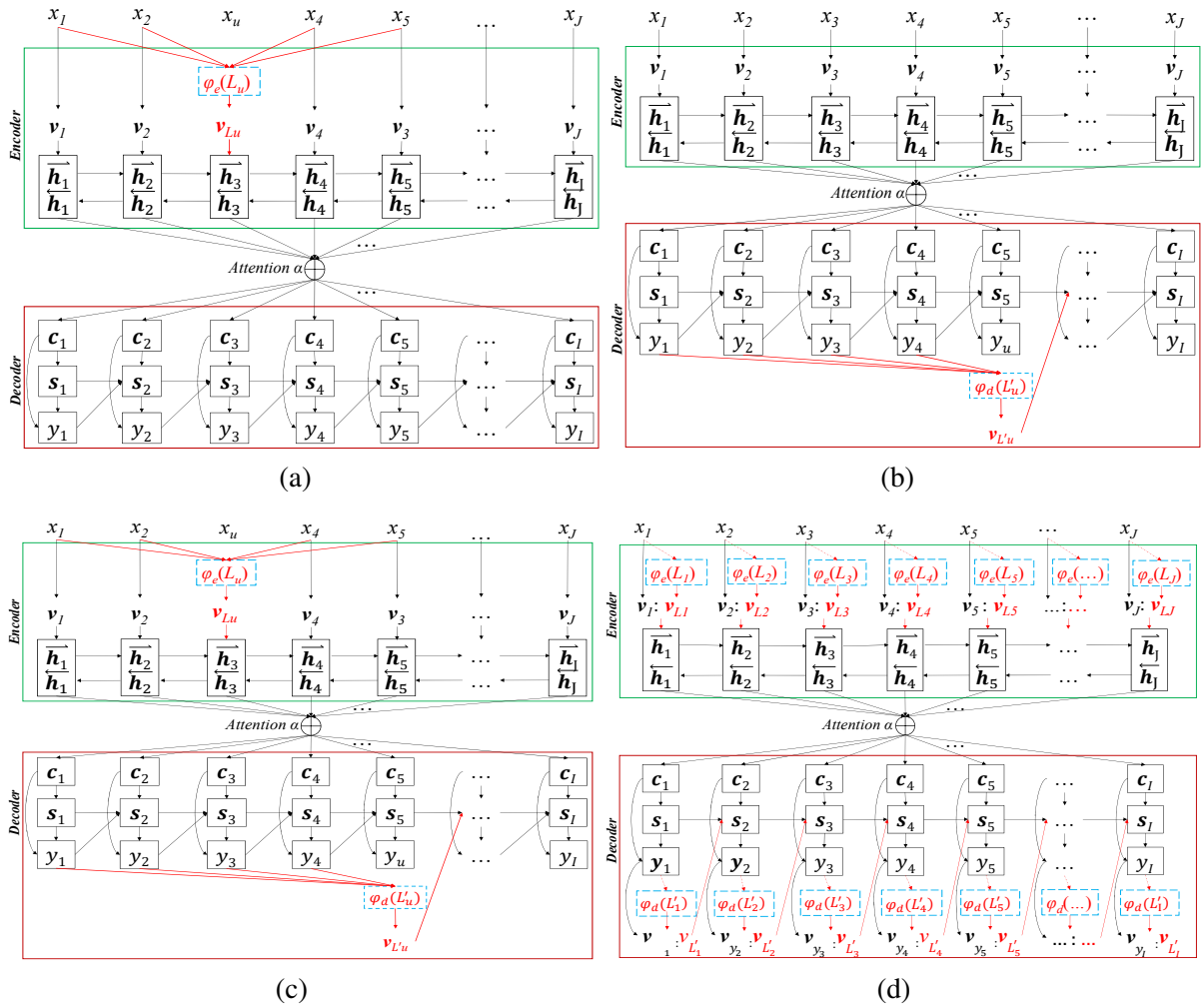

Figure 3: (a) **CARNMT**-Encoder; (b) **CARNMT**-Decoder; (c) **CARNMT**-Both; (d) **ALLSmooth**, in which the red dotted arrows obtain the context words of each word according to eq.(14) or eq.(16). The blue dotted boxes denote FCWM or CCWM proposed in section 2.

- **CARNMT**-Decoder: Only smoothing target-side unk in the *decoder*, as shown in Figure 3 (b).
- **CARNMT**-Both: Both smoothing the unks of source-side and target-side in the NMT, as shown in Figure 3 (c).
- **ALLSmooth**: this model smooths not only the *unk* words, but also all source-and target-side in-vocabulary words by the learned CARs, as shown in Figure 3 (d). Meanwhile, the vector of in-vocabularys word and its CARs are concatenated as a novel vector to represent the semantic information of the word instead of replacing the word vector with its CAR.

In our experiments, each model has two variants according to the integrated FCWM or CCWM. For example, "CARNMT-encoder (CCWM)" indicates that the CAR for OOV is learned by the CCWM proposed in the section 3. In Figure 3, we take FCWM to learn the CAR for each word (including OOV). Therefore, there is easy to use the proposed CCWM instead of the FCWM.

Moreover, the proposed NMT models with CAR are an integrative architecture without any external information. Especially, the NMT and FCWM or CCWM, which are not isolated from each other, are trained by optimizing their parameters jointly. In other words, the θ_1 or θ_2 and the parameters of NMT are optimized jointly.

5 Experiments

5.1 Setting up

We carry out experiments on the Chinese-to-English translation task. The training dataset

System	Dev (MT02)	MT03	MT04	MT05	MT06	MT08	AVG
Moses	33.15	31.02	33.78	30.33	29.62	23.53	29.66
Bahdanau et al. (2015)	36.42	34.22	37.11	33.02	32.69	25.38	32.48
Sennrich et al. (2016)	36.89	35.39	38.24	33.73	32.74	26.22	33.26
Costa-jussà and Fonollosa (2016)	35.98	34.93	37.56	33.24	32.32	26.02	32.81
Li et al. (2016)	36.96	35.78	38.42	34.02	33.14	26.36	33.54
CARNMT-Encoder (FCWM)	36.78	35.56**	38.14*	33.69	33.13	26.16*	33.34
CARNMT-Decoder (FCWM)	36.67	34.65	37.60	33.26	33.01	26.15*	32.93
CARNMT-Both (FCWM)	37.36	35.43**	38.34**	33.43	**33.47**	**26.86****	33.50
ALLSmooth (FCWM)	37.71	35.73**	**38.53****	33.91*	**33.53***	**27.18****	**33.78**
CARNMT-Encoder (CCWM)	37.12	35.64**	38.14*	33.49	33.26*	26.57**	33.42
CARNMT-Decoder (CCWM)	36.33	34.56	37.43	33.24	32.96	25.86	32.81
CARNMT-Both (CCWM)	37.56	**35.83****	**38.52****	33.73	**33.37****	**27.06****	**33.70**
ALLSmooth (CCWM)	37.69	**36.23****	**38.89****	**34.69****	**33.83****	**27.94**‡	**34.32**

Table 1: Results on NIST Chinese-to-English Translation Task. "*" indicates statistically significant better than Bahdanau et al. (2015) at p-value < 0.05 and "**" at p-value < 0.01. "†" indicate statistically significant difference (p-value < 0.05) from the Li et al. (2016) which performed the best among baselines and "‡" at p-value < 0.01. **AVG** is average BLEU scores for MT03-MT08 test sets. The bold denotes the proposed model is superior to the Li et al. (2016) over the same test set.

consists of 1.42M sentence pairs extracted from LDC corpora.[6] We choose the NIST 2002 (MT02) and the NIST 2003-2008 (MT03-08) datasets as validation set and test sets, respectively. Case-insensitive *4*-gram NIST BLEU score (Papineni et al., 2002) is as evaluation metric, and the signtest (Collins et al., 2005) was as statistical significance test.

The baseline systems included the standard PB-SMT implemented in Moses (Koehn et al., 2007) and the standard attentional NMT (Bahdanau et al., 2015) . We also compared with state-of-the-art enhanced NMT methods for OOV: subword-based NMT (Sennrich et al., 2016), character-based NMT (Costa-jussà and Fonollosa, 2016), and replacing *unk* with similarity semantic in-vocabulary words (Li et al., 2016). All of these baselines and the proposed method are implemented in Nematus [7] (Sennrich et al., 2017).

For all NMT systems, we limit the source and target vocabularies to 30K, and the maximum sentence length is 80. We shuffle training set before training and the mini-batch size is 80. The word embedding dimension is 620-dimensions [8], the hidden layer dimension is 1000, and the default dropout technique (Hinton et al., 2012) in Nematus is used on the all the layers. Training is conducted on a single Tesla P100 GPU. All NMT models trained for 15 epochs[9] using ADADELTA optimizer (Zeiler, 2012), and our training time is only about 10% slower than the standard attentional NMT.

5.2 Results and Analyses

Table 1 shows the translation performances on test sets measured in BLEU score. The standard attentional NMT (Bahdanau et al., 2015) outperforms Moses by 2.78 BLEU points on average, indicating that it is a strong baseline NMT system. All the comparison methods, including Sennrich et al. (2016), Costa-jussà and Fonollosa (2016), and Li et al. (2016), outperform the standard attentional NMT.

1) Over the standard attentional NMT, CARNMT-Encoder (FCWM/CCWM) gain improvements of 0.86/0.94 BLEU points on average, and CARNMT-Decoder (FCWM/CCWM) gain improvements of 0.45/0.33 BLEU points on average. CARNMT-Both (FCWM/CCWM) gain improvements of 1.02/1.30 BLEU points on average, which indicates that improvement in encoder and decoder are essentially orthogonal.

2) ALLSmooth (FCWM/CCWM) surpass CARNMT-Both (FCWM/CCWM) by 0.28/0.62 BLEU points on average. This indicates that the proposed context-aware smoothing method not only helps relieve the OOV affect, but also enhances representations of in-vocabulary words.

3) ALLSmooth (FCWM/CCWM) also outperforms the best performed baseline Li et al. (2016), which replaces the *unk* words by using external lexicon similarity, by 0.24/0.78 BLEU points on

[6] LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08, and LDC2005T06.

[7] https://github.com/EdinburghNLP/nematus

[8] For the ALLSmooth, the 360 dimensions are from V_{x_j} or V_{y_i} and the 260 dimensions were from the learned CAR

[9] All NMT models are convergent in the 15 epochs.

SRC: 用好 这个 战略 机遇期 (OOV) , 力争 有所 作为 , 必须 把 发展 科学 技术 放在 更加 重要 , 更加 突出 的 位置
(***pinyin***) *yonghao zhege zhanlue jiyuqi , lizheng yousuo zuowei , bixu ba fazhan kexue jishu fangzai gengjia zhongyao , gengjia tuchu de wiezhi*

Bahdanauet al.(2015) : to make good use of this strategy , we should strive for the development of science and technology , and must put the development of science and technology into an even more important and prominent position

This work: in making good use of this strategic plan and striving to accomplish something , it is necessary to place the development of science and technology in a more important and more prominent position

Ref: to well use this strategic period of opportunity and strive to accomplish some achievments , the development of science and technology should be placed in a more prior and prominent position

Figure 4: Translation sample for source sentence with one OOV. The English phrases in color indicate they are translations from the corresponding Chinese phrase with the same color.

average.

4) The CCWM performs slightly better than FCWM. The reason may be that the convolution neural network can summarize the contextual information better than the feedforward neural network.

5.3 Translation Qualities for Sentences with Different Numbers of OOV

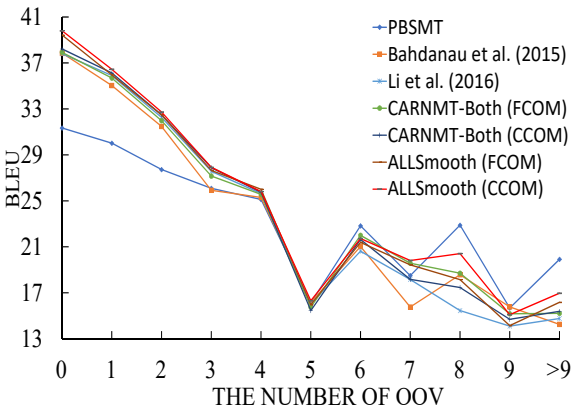

Figure 5: Translation qualities for sentences with different numbers of OOV.

To further verify our methods, we group sentences of same number OOVs all the test sets (MT03-08), for example, "5" indicates that all the source sentences include five OOV words in the group, and compute a BLEU score per group.

1) In Figure 5, we observe that when the number of OOVs is zero (no OOV), ALLSmooth (FCWM/CCWM) outperform other baseline systems, and the performances of CARNMT-Both (FCWM/CCWM) are similar to standard attentional NMT. This means that CARNMT-Both (FCWM/CCWM) degrade into standard attentional NMT because of these sentences not include OOV, but our context-aware smoothing method enhances the representation of in-vocabulary words in the ALLSmooth (FCWM/CCWM).

2) With the increasing in the number of OOVs (especially when more than five), the gap between our methods and other methods (except PBSMT) become larger. This indicates that our methods are especially good at dealing with multi-OOV situation, in comparison with other NMT methods.

5.4 Samples Analysis

This subsection shows one translation sample for source sentence with one OOV, as shown in Figure 4. We compare our method ALLSmooth (CCWM) with Bahdanau et al. (2015) on the translation of a source sentence with the OOV "*jiyuqi*" ("*period of opportunity*" in English).

1) For both of Bahdanau et al. (2015)'s method and the proposed method, the OOV "*jiyuqi*" itself is not translated.

2) For Bahdanau et al. (2015)'s method, the phrase "*lizheng yousuo zuowei*" ("*strive to accomplish some achievments*" in English) after "*jiyuqi*" is not translated. The purple part of source sentence are translated twice in (Bahdanau et al., 2015)'s method. This is in consistent with our hypothesis in Section 1: the OOV which makes the structure of source sentence discontinuous affects source representation learning in encoder; then the negative effect would be further transformed to the decoder by the source annotation vectors, thus generating the poverty context vector and hidden layer state for translation prediction.

3) In comparison, the proposed method translates it into "*striving to accomplish something*", which is quite close to the reference. This indicates that our proposed context-aware smoothing method can relieve more translation

performance for NMT from the OOV's negative effect shown in Section 1.

6 Conclusion

In this paper, we explored the context information to smooth source representation with OOVs, and integrate the learned CAR into the Encoder and Decoder of NMT to improve the translation performance. Especially, we extended the method to smooth each word in-vocabulary, and further gained improvements over the proposed models for the NMT.

In the future, we will exploit richer context information, such as pos-tagger and named entity, to enhance the semantic representation of vocabulary in NMT.

Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments and suggestions. This work was partially supported by the program "Promotion of Global Communications Plan: Research, Development, and Social Demonstration of Multilingual Speech Translation Technology" of MIC, Japan. Tiejun Zhao is supported by the National Natural Science Foundation of China (NSFC) via grant 91520204 and National High Technology Research & Development Program of China (863 program) via grant 2015AA015405.

References

Yaser Al-Onaizan and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 400–408, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Austin, Texas. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 531–540, Ann Arbor, Michigan. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 357–361, Berlin, Germany. Association for Computational Linguistics.

Pascale Fung and Percy Cheung. 2004. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and em. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 57–63, Barcelona, Spain. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

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*, pages 1–10, Beijing, China. Association for Computational Linguistics.

Long Jiang, Ming Zhou, Lee-Feng Chien, and Cheng Niu. 2007. Named entity translation with web mining and transliteration. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 1629–1634, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Kevin Knight and Jonathan Graehl. 1997. Machine transliteration. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Madrid, Spain. Association for Computational Linguistics.

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. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2016. Towards zero unknown word in neural machine translation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2852–2858.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1054–1063, Berlin, Germany. Association for Computational Linguistics.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in 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*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 381–390, Singapore. Association for Computational Linguistics.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Vocabulary manipulation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 124–129, Berlin, Germany. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Shachar Mirkin, Lucia Specia, Nicola Cancedda, Ido Dagan, Marc Dymetman, and Idan Szpektor. 2009. Source-language entailment modeling for translating unknown terms. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 791–799, Suntec, Singapore. 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 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Li Shao and Hwee Tou Ng. 2004. Mining new word translations from comparable corpora. In *Proceedings of COLING 2004, the 20th International Conference on Computational Linguistics*, pages 618–624, Geneva, Switzerland. COLING.

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.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701.