

Reference Network for Neural Machine Translation

Han Fu^{†‡} Chenghao Liu[§] Jianling Sun^{†‡*}

[†]Zhejiang University, Hangzhou, China

[‡]Alibaba-Zhejiang University Joint Institute of Frontier Technologies, China

[§]Singapore Management University, Singapore

{11821003, sunjl}@zju.edu.cn

twinsken@gmail.com

Abstract

Neural Machine Translation (NMT) has achieved notable success in recent years. Such a framework usually generates translations in isolation. In contrast, human translators often refer to reference data, either rephrasing the intricate sentence fragments with common terms in source language, or just accessing to the golden translation directly. In this paper, we propose a Reference Network to incorporate referring process into translation decoding of NMT. To construct a *reference book*, an intuitive way is to store the detailed translation history with extra memory, which is computationally expensive. Instead, we employ Local Coordinates Coding (LCC) to obtain global context vectors containing monolingual and bilingual contextual information for NMT decoding. Experimental results on Chinese-English and English-German tasks demonstrate that our proposed model is effective in improving the translation quality with lightweight computation cost.

1 Introduction

Neural Machine Translation (NMT) has enjoyed impressive success in most large-scale translation tasks (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014). Typical NMT model to date is a single end-to-end trained deep neural network that encodes the source sentence into a fixed-length vector and generates the words in the target sentence sequentially. The alignment relationship between source and target sentence is learned by the attention mechanism (Bahdanau et al., 2015; Luong et al., 2015).

Though the framework has achieved significant success, one critical concern is that NMT generates translations in isolation, which leads to translation inconsistency and ambiguity arising from

a single source sentence (Tu et al., 2018). Recently, there have been few attempts to model the semantic information across sentences. The basic ideas are to store a handful of previous source or target sentences with context vectors (Jean et al., 2017; Wang et al., 2017a) or memory components (Maruf and Haffari, 2018; Tu et al., 2018). However, these methods have several limitations. First, the very short view of the previous sentences (usually one or two sentence(s)) is not sufficient enough to catch long term dependencies across paragraphs and storing detailed translation history is computationally expensive. Second, in the real-world scenarios, input data of MT application is often isolated sentences, such as *Google Translate*, where no cross-sentence contexts are provided. Moreover, translations generated by such document-level NMT models are not stable, effected by the sentences surrounding the current one to translate.

To address these limitations, we model the semantic information across sentences by mimicking the human translation process. In real scenarios, there will always be sentences or fragments that the translator can understand the meaning but cannot write down the translations directly. The obstacle could be unfamiliar collocation, descriptions in specific language habits and slang. The usual solutions for human are: (1) paraphrasing the sentence in another way, with simpler and more colloquial terms in the source language, and (2) directly referring to the standard translations of the intricate sentence fragments. For example in Table 1, the Chinese word "zaiyu" is not a common expression. A reference can either provide simple Chinese terms such as "daizhe rongyu" or directly offer the corresponding English translation "with honor". Therefore, if a good quality *reference book* which covers various translation scenes is provided, it can definitely improve the

*Corresponding author: Jianling Sun.

source	canjia dongaohui de faguo yun-dongyuan zaiyu fanhui bali.
translation	French athletes participating in winter olympics returned to paris with honors .

Table 1: An example of sentence fragment that is hard to translate.

performance of human translators.

To be specific, the motivation of this work can be summarized as two aspects corresponding to the two kinds of human reference processes. First, we aim to provide the machine translator with a reference during decoding, which contains all possible source sentence fragments that are semantically similar to the current one. If the system finds it hard to translate the source fragment, it can turn to translate the fragments in the reference. Second, we intend to offer the oracle translations of the current sentence fragments to translate.

In this paper, we propose a novel model namely Reference Network that incorporates the referring process into translation decoding of NMT. Instead of storing the detailed sentences or translation history, we propose to generate representations containing global monolingual and bilingual contextual information with Local Coordinate Coding (LCC) (Yu et al., 2009). Specifically, for solution (1), the hidden states of NMT encoder are coded by a linear combination of a set of anchor points in an unsupervised manner. The anchors are capable to cover the entire latent space of the source language seamlessly. For solution (2), we employ local codings to approximate the mapping from source and target contexts to the current target word with a supervised regression function. The local coding is then fed to the decoder to modify the update of the decoder hidden state. In this way, the translation decoding can be improved by offering the representation of a common paraphrase (Figure 1) or golden target translation (Figure 2).

We conduct experiments on NIST Chinese-English (Zh-En) and WMT German-Chinese (En-De) translation tasks. The experimental results indicate that the proposed method can effectively exploit the global information and improve the translation quality. The two proposed models significantly outperform the strong NMT baselines by adding only 9.3% and 19.6% parameters respectively.

2 Background

2.1 Neural Machine Translation

Our model is built on the RNN-based NMT (Bahdanau et al., 2015). However, since recurrent architecture is not necessary for our approach, the idea can also be applied to ConvS2S (Gehring et al., 2017) and Transformer (Vaswani et al., 2017). We leave it for future work. Formally, let $\mathbf{x} = (x_1, \dots, x_m)$ be a given source sentence and $\mathbf{y} = (y_1, \dots, y_T)$ be the corresponding target sentence. NMT generates the target words sequentially by maximizing the probability of translation conditioned on the source sentence:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} \sum_{t=1}^T \log p(y_t | \mathbf{x}, \mathbf{y}_{<t}). \quad (1)$$

At each timestep, the generation probability is computed as

$$p(y_t | \mathbf{x}, \mathbf{y}_{<t}) = \text{softmax}(g(\mathbf{e}(y_{t-1}), \mathbf{s}_t, \mathbf{c}_t)), \quad (2)$$

where g is a transformation function that outputs a vocabulary-sized vector, $\mathbf{e}(y_{t-1})$ is the embedding of previous target word y_{t-1} , \mathbf{c}_t is the source context vector obtained by attention mechanism, and \mathbf{s}_t is the t -th hidden state of NMT decoder, computed as:

$$\mathbf{s}_t = f_d(\mathbf{e}(y_{t-1}), \mathbf{s}_{t-1}, \mathbf{c}_t), \quad (3)$$

where f_d is a nonlinear activation. The source context \mathbf{c}_t is typically a weighted sum of encoder hidden states as:

$$\mathbf{c}_t = \sum_{i=1}^m \alpha_{ti} \cdot \mathbf{h}_i, \quad (4)$$

where attention score α_{ti} is the alignment vector of the i -th source word x_i and the t -th target word y_t :

$$\alpha_{ti} = \text{softmax}(\mathbf{v}_\alpha^\top \tanh(\mathbf{W}_\alpha \mathbf{s}_{t-1} + \mathbf{U}_\alpha \mathbf{h}_i)). \quad (5)$$

where \mathbf{W}_α , \mathbf{U}_α and \mathbf{v}_α are trainable matrices or vectors. \mathbf{h}_i is the annotation of x_i computed by the NMT encoder. The encoder, generally implemented as a bi-directional RNN, encodes the input sentence into a sequence of source hidden states $\mathbf{h} = (\mathbf{h}_1, \dots, \mathbf{h}_m)$ where \mathbf{h}_i is obtained by concatenating the forward hidden state $\overrightarrow{\mathbf{h}}_i$ and backward one $\overleftarrow{\mathbf{h}}_i$ at timestep i .

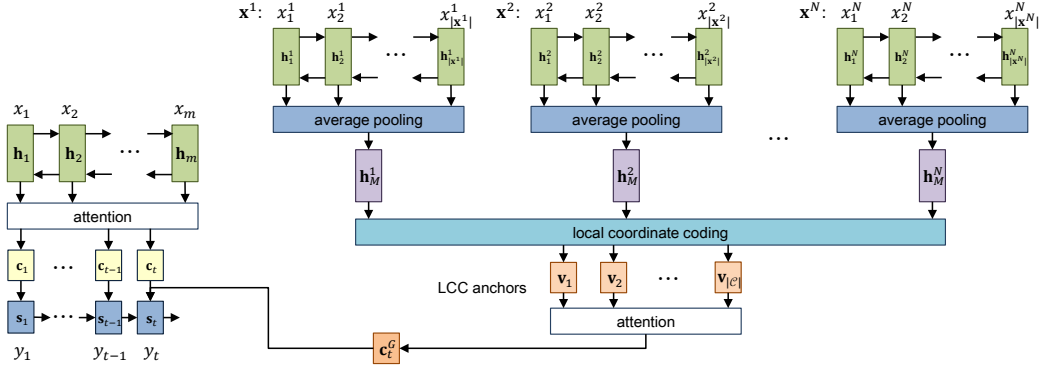


Figure 1: Framework of NMT with M-RefNet. \mathbf{x}^i represents the i -th source sentence in the training corpus and $|\mathbf{x}^i|$ is the length of the sentence. The global context vector \mathbf{c}_t^G can be regarded as a paraphrase of the current source context \mathbf{c}_t .

According to the above formulations, conventional NMT models translate sentences independently. However, human translators usually tend to seek for reference materials when in trouble. Motivated by such common human behaviors, we propose Reference Network to provide global information as a *reference book* in two ways. First, the model utilizes all source hidden states to paraphrase current source sentence. Second, the model directly provides the target word \tilde{y}_t according to the rest translation samples in the training corpus. Since it is impossible to store all information directly, we leverage local coordinate coding (LCC) to compress the semantics into a latent manifold.

2.2 Local Coordinate Coding

With the assumption that data usually lies on the lower dimensional manifold of the input space, the manifold approximation of high dimensional input \mathbf{x} can be defined as a linear combination of surrounding anchor points as:

$$\mathbf{x} \approx \gamma(\mathbf{x}) = \sum_{\mathbf{v}} \gamma_{\mathbf{v}}(\mathbf{x}) \mathbf{v}, \quad (6)$$

where \mathbf{v} is an anchor point and $\gamma_{\mathbf{v}}$ is the weight corresponding to \mathbf{v} such that

$$\sum_{\mathbf{v}} \gamma_{\mathbf{v}}(\mathbf{x}) = 1. \quad (7)$$

According to the definitions, it is proved in (Yu et al., 2009) that if the anchor points are localized enough, any (l_α, l_β) -Lipschitz smooth function $f(\mathbf{x})$ defined on a lower dimensional manifold \mathcal{M} can be globally approximated by a linear

combination of the function values of a set of the anchors \mathcal{C} as:

$$f(\mathbf{x}) \approx \sum_{\mathbf{v} \in \mathcal{C}} \gamma_{\mathbf{v}}(\mathbf{x}) f(\mathbf{v}), \quad (8)$$

with the upper bound of the approximation error:

$$l_\alpha \|\mathbf{x} - \sum_{\mathbf{v} \in \mathcal{C}} \gamma_{\mathbf{v}}(\mathbf{x}) \mathbf{v}\| + \sum_{\mathbf{v} \in \mathcal{C}} l_\beta |\gamma_{\mathbf{v}}(\mathbf{x})| \|\mathbf{v} - \sum_{\mathbf{v} \in \mathcal{C}} \gamma_{\mathbf{v}}(\mathbf{x}) \mathbf{v}\|^2. \quad (9)$$

3 Reference Network

In this section, we present our proposed Reference Network (RefNet).

3.1 Overview

We propose two models which explore the global information from the training data in different manners as illustrated by Figure 1 and Figure 2.

The monolingual reference network (M-RefNet) provides a global source context vector to paraphrase the current context \mathbf{c}_t based on all other source sentences. To be specific, we train several unsupervised anchors as the bases of the semantic space of source contexts and each source sentence in the training corpus can be represented by a weighted sum of the anchors.

The bilingual reference network (B-RefNet) generates a referable target embedding according to all sentence pairs in the training corpus to guide output sequence generation. Concretely, we formulate the translation process as a mapping from source and target contexts (\mathbf{c}_t and \mathbf{s}_{t-1}) to the current target word embedding $\mathbf{e}(y_t)$. B-RefNet

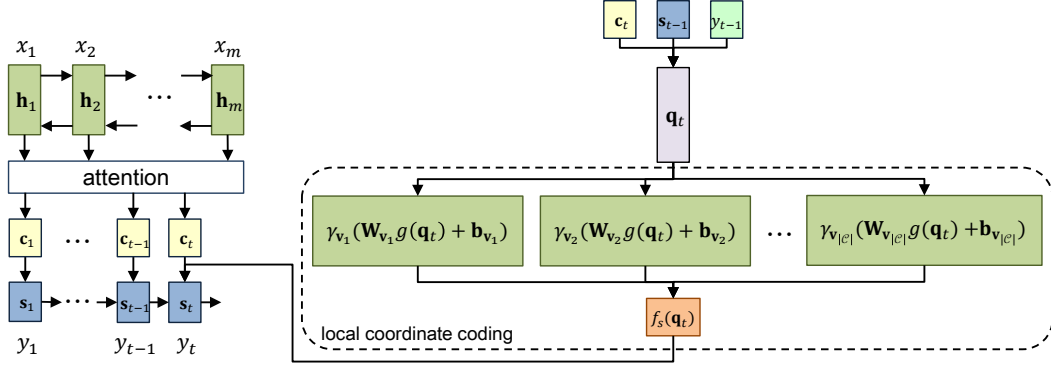


Figure 2: Framework of NMT with B-RefNet. The output $f_s(\mathbf{q}_t)$ of RefNet can be regarded as an approximation of current target word embedding $\mathbf{e}(y_t)$.

learns this mapping with a supervised regression function derived from LCC.

It should be noted that the corpus from which the reference vectors (\mathbf{c}_t^G or $f_s(\mathbf{q}_t)$) are learned can be any monolingual or bilingual data, and the translations generated are relatively effected by the quality of the corpus. In this work, we constrain it as the training corpus for convenience and a fair comparison with the related work.

3.2 Monolingual Referent Network

In this section, we seek to improve NMT by rephrasing the source sentence. Instead of storing all source contexts, we regenerate the source contexts from a learned manifold with a combination of a fixed number of anchor points. Formally, given any source sequence \mathbf{x} with length m in the training samples, let $\mathbf{h} = (\mathbf{h}_1, \dots, \mathbf{h}_m)$ denotes the hidden states generated by the NMT encoder. We firstly obtain the representation of the source sentence \mathbf{h}_M via a *mean-pooling* operation. According to the definition of LCC, it can be assumed that $\mathbf{h}_M \approx \gamma(\mathbf{h}_M)$ where $\gamma(\mathbf{h}_M)$ is the local coordinate coding of \mathbf{h}_M , computed as:

$$\gamma(\mathbf{h}_M) = \sum_{j=1}^{|\mathcal{C}|} \gamma_j(\mathbf{h}_M) \mathbf{v}_j. \quad (10)$$

Here, \mathbf{v}_j is the j -th anchor point. The coefficient $\gamma_j(\mathbf{h}_M)$ is used to measure the weight of anchor point \mathbf{v}_j corresponding to $\gamma(\mathbf{h}_M)$. In conventional manifold learning methods, $\gamma_j(\mathbf{h}_M)$ is generally computed with distance measure. And to achieve localization, the coefficients corresponding to anchor points out of the neighbors of \mathbf{h}_M are set to zero. However, it is hard to train in deep

neural network using stochastic gradient methods. Inspired by the attention mechanism (Bahdanau et al., 2015), we propose to employ an attention layer to obtain the weights:

$$\gamma_j(\mathbf{h}_M) = \frac{\exp(s(\mathbf{h}_M, \mathbf{v}_j))}{\sum_{j=1}^{|\mathcal{C}|} \exp(s(\mathbf{h}_M, \mathbf{v}_j))}, \quad (11)$$

where $s(\cdot)$ is a score function. Here, we propose a tri-nonlinear score function which has been proven especially effective in the experiments:

$$s(\mathbf{h}_M, \mathbf{v}_j) = \mathbf{v}_s^\top \tanh(\mathbf{W}_s \mathbf{v}_j + \mathbf{U}_s \mathbf{h}_M) + \mathbf{V}_s(\mathbf{v}_j \circ \mathbf{h}_M), \quad (12)$$

where \mathbf{W}_s , \mathbf{U}_s , \mathbf{V}_s and \mathbf{v}_s are trainable parameters. \circ is the element-wise multiplication, and dimension of any anchor point should be the same to \mathbf{h}_M .

To find the optimal anchor point, *localization measure* (Yu et al., 2009) is employed as the optimization object:

$$\min_{\gamma, \mathcal{C}} l_\alpha \|\mathbf{h}_M - \gamma(\mathbf{h}_M)\| + l_\beta \sum_{j=1}^{|\mathcal{C}|} |\gamma_j(\mathbf{h}_M)| \|\mathbf{v}_j - \gamma(\mathbf{h}_M)\|^2. \quad (13)$$

Since any source sentence presentation \mathbf{h}_M can be represented by the linear combination of the anchors, the trained anchor points can be regarded as the bases of the latent space of all source annotations, containing the global contextual information. Therefore, during translation decoding of NMT, we can drop the coefficient γ and rephrase the source sentence only with the anchor points.

Specifically, we apply an attention mechanism between current local contextual information and each anchor point \mathbf{v}_j to get the global context as:

$$\mathbf{c}_t^G = \sum_{j=1}^{|\mathcal{C}|} \alpha_{tj}^G \mathbf{v}_j, \quad (14)$$

where α_{tj}^G is the attention score between current local contexts and the global context, computed as:

$$\alpha_{tj}^G = \text{softmax}(\mathbf{v}_\alpha^\top \tanh(\mathbf{W}_\alpha \mathbf{s}_{t-1} + \mathbf{U}_\alpha \mathbf{c}_t + \mathbf{V}_\alpha \mathbf{v}_j)). \quad (15)$$

Once the global context \mathbf{c}_t^G is obtained, we feed it to decoder states:

$$\mathbf{s}_t = f_d(\mathbf{e}(y_{t-1}), \mathbf{s}_{t-1}, \mathbf{c}_t, \mathbf{c}_t^G), \quad (16)$$

where \mathbf{c}_t encodes the local contextual information and \mathbf{c}_t^G contains the global monolingual information from all source sentences in the training corpus. When the model has trouble to translate some words or sentence fragments, it can refer to \mathbf{c}_t^G to gain the richer source contextual information.

3.3 Bilingual Reference Network

The bilingual model is proposed to improve NMT by providing a golden translations according to rest samples in the training corpora. To be specific, once source context \mathbf{c}_t and target context \mathbf{s}_{t-1} are obtained, we hope to provide a referable prediction $\mathbf{e}(\hat{y}_t)$ of the current target word embedding $\mathbf{e}(y_t)$ according to other sentence pairs in the training data for the decoder.

The functionality of the NMT decoder during translation (Eq.2 and Eq.3) is totally a function that maps the source context \mathbf{c}_t , target context \mathbf{s}_{t-1} and last target word y_{t-1} to current target y_t . NMT takes it as a classification problem, using tanh or other gated RNN unit to implement this function. In this work, we propose a much stronger model in information expression, that regrades the problem as regression:

$$\mathbf{q}_t = [\mathbf{e}(y_{t-1})^\top, \mathbf{s}_{t-1}^\top, \mathbf{c}_t^\top]^\top, \quad (17)$$

$$\mathbf{e}(y_t) \approx f_s(\mathbf{q}_t) = \mathbf{W}(\mathbf{q}_t)g(\mathbf{q}_t) + \mathbf{b}(\mathbf{q}_t), \quad (18)$$

where g is a transformation function that transforms \mathbf{q}_t to a anchor-size vector, \mathbf{W} and \mathbf{b} are the weight matrix and bias vector of the regression function. The weight and bias are allowed to

vary according to the input \mathbf{q}_t , which makes the function capable of mapping each \mathbf{q}_t to the corresponding $\mathbf{e}(y_t)$ precisely. However, it is impossible to store the weight and bias for every \mathbf{q}_t computed within the training data. Therefore, we approximate the weight and bias function in Eq.18 using local coordinate coding as:

$$f_s(\mathbf{q}_t) = \sum_{j=1}^{|\mathcal{C}|} \gamma_j(\mathbf{q}_t) (\mathbf{W}_{\mathbf{v}_j} g(\mathbf{q}_t) + \mathbf{b}_{\mathbf{v}_j}), \quad (19)$$

where $\mathbf{v}_j \in \mathcal{C}$ is an anchor point, $\mathbf{W}_{\mathbf{v}_j}$ and $\mathbf{b}_{\mathbf{v}_j}$ are trainable parameters corresponding to \mathbf{v}_j , and $\gamma_j(\mathbf{q}_t)$ is the weight function, computed as:

$$\gamma_j(\mathbf{q}_t) = \frac{\exp(s(\mathbf{q}_t, \mathbf{v}_j))}{\sum_{j=1}^{|\mathcal{C}|} \exp(s(\mathbf{q}_t, \mathbf{v}_j))}. \quad (20)$$

Similar to M-RefNet, the score is computed by the tri-nonlinear function as:

$$s(\mathbf{q}_t, \mathbf{v}_j) = \mathbf{v}_b^\top \tanh(\mathbf{W}_b \mathbf{v}_j + \mathbf{U}_b \mathbf{q}_t + \mathbf{V}_b (\mathbf{v}_j \circ \mathbf{q}_t)). \quad (21)$$

Here, $f_s(\mathbf{q}_t)$ can be regarded as an approximation of $\mathbf{e}(y_t)$ based on all the sentence pairs in the training data. Therefore, we feed the function value to the decoder state to guide sentence generation:

$$\mathbf{s}_t = f_d(\mathbf{e}(y_{t-1}), \mathbf{s}_{t-1}, \mathbf{c}_t, f_s(\mathbf{q}_t)). \quad (22)$$

The optimal weight matrices and anchor points are obtained by minimizing the hinge loss for each sentence pair (\mathbf{x}, \mathbf{y}) as:

$$\mathcal{L}_M = \sum_{t=1}^{|\mathcal{Y}|} \|\mathbf{e}(y_t) - f_s(\mathbf{q}_t)\|^2 + \lambda_M \sum_{j=1}^{|\mathcal{C}|} \|\mathbf{W}(\mathbf{v}_j)\|^2. \quad (23)$$

3.4 Training and Inference

Stage-wise training strategies have been proven to be efficient when system is relative complicated by plenty of recent work (Maruf and Haffari, 2018; Tu et al., 2018). In this work, we first pre-train a standard NMT on a set of training examples $\{\{\mathbf{x}^n, \mathbf{y}^n\}\}_{n=1}^N$ as initialization for training the added parameters in our proposed models.

Let $\theta = \{\theta_E, \theta_D\}$ denote the parameters of the standard NMT, where θ_E and θ_D are parameters of the standard encoder and decoder (including attention model) respectively. For M-RefNet, the stage following NMT training is to obtain the weight vectors γ and anchor points \mathcal{C} related to all training

System	MT05	MT06	MT08	Avg
DI4mt	32.88	32.30	25.97	30.38
NMT	35.76	34.82	27.86	32.81
CS-NMT	36.63	36.41	29.47	34.17
LS-NMT	36.46	36.99	29.73	34.39
CC-NMT	36.65	37.08	29.71	34.48
DC-NMT	36.82	36.73	29.83	34.46
<i>This work</i>				
M-RefNet	37.31	37.72	30.41	35.15
B-RefNet	37.71	37.99	30.80	35.50

Table 2: BLEU scores of different models on Zh-En.

sentence representations \mathbf{h}_M by minimizing *localization measure* (Eq.13). Then we fix the trained anchor points and encoder, and only fine-tune the decoder θ_D and the added parameters θ_M related to the monolingual reference network (Eq.15 and Eq.16):

$$\max_{\theta_D, \theta_M} \sum_{n=1}^N [\log P(\mathbf{y}^n | \mathbf{x}^n; \theta, \theta_M, \gamma)]. \quad (24)$$

To train B-RefNet efficiently, we fix the trained parameters of the standard NMT and only update the added parameters θ_B including all weight matrices and biases related to local coordinate coding (Eq.19 and Eq.21). The training object is:

$$\max_{\theta_B} \sum_{n=1}^N [\log P(\mathbf{y}^n | \mathbf{x}^n; \theta, \theta_B) - \lambda \mathcal{L}_M], \quad (25)$$

where λ is a hyper-parameter that balances the preference between likelihood and hinge loss.

During inference, all parameters related to LCC are fixed. Therefore, the work can be regarded as a *static* approach, compared with the conventional document-level NMT. That means, the final translation is only effected by the reference corpus but not by the sentences surrounding the current one to translate. Naturally, there leaves a question that how it influences the quality of translations when various reference corpus is chosen. We leave it in future work and only use the training corpus in this paper.

4 Experiments

We evaluate the reference network models on two translation tasks, NIST Chinese-English translation (Zh-En) and WMT English-German translation (En-De).

4.1 Settings

Datasets For Zh-En, we choose 1.25M sentence pairs from LDC dataset¹ with 34.5 English words and 27.9M Chinese words. NIST MT02 is chosen as the development set, and NIST MT05/06/08 as test sets. Sentences with more than 50 words are filtered and vocabulary size is limited as $30k$. We use case-insensitive BLEU score to evaluate Zh-En translation performance. For En-De, the training set is from (Luong et al., 2015) which contains 4.5M bilingual pairs with 116M English words and 100M German words. BPE (Sennrich et al., 2016) is employed to split the sentence pairs into subwords and we limit the vocabulary as $40k$ sub-words units. Newstest2012/2013 are chosen for developing and Newstest2014 for test. case-sensitive BLEU² is employed as the evaluation metric.

Models We evaluate our RefNet with different structures on Zh-En and En-De. For Zh-En we choose the typical attention-based recurrent NMT model (Bahdanau et al., 2015) as initialization, which consists of a bi-directional RNN-based encoder and a one layer RNN decoder. The dimensions of embedding and hidden state are 620 and 1000 respectively. For En-De, deep linear associative unit model (DeepLAU) (Wang et al., 2017b) is chosen as the base model. Both the encoder and decoder consist of 4-layer LAUs. All embedding and hidden states are 512-dimensional vectors. Moreover, we use layer normalization (Ba et al., 2016) on all layers. For both architectures, the number of anchor points is 100 for M-RefNet and 30 for B-RefNet. The anchor dimension of B-RefNet is set to 100. The hyper-parameter λ in Eq.25 is set to 1. The norm of gradient is clipped to be within $[-1, 1]$ and dropout is applied to embedding and output layer with rate 0.2 and 0.3 respectively. When generating translations, we utilize beam search with beam size 10 on Zh-En and 8 on En-De.

4.2 Results on Chinese-English Translation

The standard attention-based NMT model is chosen as the baseline and initialization of our models. Moreover, we also list the results of the open-

¹The corpus contains LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06

²<https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl>

source D14mt and re-implementations of the following related work for comparison:

- Cross-sentence context-aware NMT (CS-NMT) (Wang et al., 2017a): A cross-sentence NMT model that incorporates the historical representation of three previous sentences into decoder.
- LC-NMT (Jean et al., 2017): A NMT model that concurrently encodes the previous and current source sentences as context, added to decoder states.
- NMT augmented with a continuous cache (CC-NMT) (Tu et al., 2018): A NMT model armed with a cache³ which stores the recent translation history.
- Document Context NMT with Memory Networks (DC-NMT) (Maruf and Haffari, 2018): A document-level NMT model that stores all source and target sentence representations of a document to guide translation generating⁴.

All the re-implemented systems share the same settings with ours for fair comparisons.

4.2.1 Main Results

Results on Zh-En are shown in Table 2. The baseline NMT significantly outperforms the open-source D14mt by 2.43 BLEU points, indicating the baseline is strong. Our proposed M-RefNet and B-RefNet improve the baseline NMT by 2.34 and 2.69 BLEU respectively and up to 2.90 and 3.17 BLEU on NIST MT06, which confirms the effectiveness of our proposed reference networks. Overall, B-RefNet achieves the best performance over all test sets

Compared with the related work which incorporate document-level information NMT, our proposed models still have a significant advantage. Compared to the best performance achieved by the related work (CC-NMT), M-RefNet and B-RefNet outperform it over all test sets and gain improvements of 0.77 BLEU and 1.02 BLEU in average. The possible reason is that all the related work only leverage a small range of the document-level information, limited by model complexity

³Cache size is set to 25.

⁴LDC training corpora contains nature boundaries. However document range is not clear for NIST test data. We use clustering and regard each class as a document. Dimension of document context is set to 1024.

#	System	#Para	Speed	
			Train	Test
0	NMT	71.1M	3590.4	114.21
1	CS-NMT	95.7M	747.5	97.10
2	LC-NMT	96.8M	1983.5	70.11
3	CC-NMT	75.1M	2844.7	113.09
4	DC-NMT	86.2M	2093.6	54.07
5	M-RefNet	77.7M	2563.98	113.26
6	B-RefNet	85.1M	2191.4	104.07

Table 3: Statistics of parameters, training speed (sentences/minute) and testing speed (words/second).

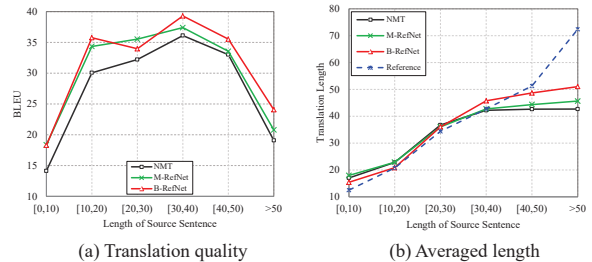


Figure 3: Translation quality and averaged length of the translations as source sentences become longer.

and time consuming. In contrast, our models are capable to express all information with more abstract representations. According to the results, though the information is deeply compressed in our models, it is still effective.

4.2.2 Analysis

Parameters and Speed The number of parameters and speed of each model are listed in Table 3. It can be seen that M-RefNet only introduces 6.6M additional parameters while B-RefNet introduces relative larger number of parameters (14M). Considering training process, both M-RefNet and B-RefNet are quite efficient and the training speeds are little slower than CC-NMT, for the added amount of parameters is quite small compared to the baseline NMT and related systems. In terms of decoding, both proposed models do not slow down the translation speed obviously and M-RefNet achieves the fastest speed over all systems except the baseline NMT. The reason is that our models do not incorporate additional previous sentences or interact with extra memory as the relevant document-level systems. Furthermore, though the training speed and number of parameters of B-RefNet and DC-NMT are similar, B-RefNet gains a twice faster translation speed, because that DC-NMT needs a two-pass translation

Source	agenting zantong yige zhongguo de lichang .
Reference	argentina supports the "one china" policy .
NMT	argentina agrees with china ' s <i>stand on</i> the one china .
M-RefNet	the argentine government supports the one china position .
B-RefNet	argentina supports the one china policy .
Source	yindu bianfang minbing 2 yue 17 ri , jiaqiang le dui niboer bianjie de xunluo .
Reference	on february 17 , the indian border security force stepped up patrols along the border with nepal .
NMT	on 17 february , indian border defense militia [<i>UNK UNK UNK UNK UNK</i>] .
M-RefNet	the indian border defense militia , on 17 february , strengthened the patrol of nepal ' s border .
B-RefNet	the indian border defense militia has stepped up patrols on the nepalese border on 17 february .

Table 4: Comparison on translation examples. The translation errors are highlighted with italic and the correct ones are highlighted with bold type.

process to fill the memory cells.

Length Analysis We follow (Luong et al., 2015) to group sentences with similar lengths and compute the BLEU score of each group, as shown in Figure 3. The reason for the falling of BLEU in the last group (>50) is that sentences longer than 50 are removed during training. From this figure, we can see that our proposed models outperform the baseline NMT in all ranges of length. Moreover, translations generated by M-RefNet and B-RefNet have more similar lengths to the references compared with the baseline NMT.

Case Study Table 4 shows the translation examples on Zh-En. In the first case, the Chinese word "lichang" (standpoint, position, or policy) is incorrectly interpreted as "stand on" by NMT. Both M-RefNet and B-RefNet generate legible translations while translation from B-RefNet is more precise. This is because the word pair ("lichang", "policy") appear somewhere in the training data and is leveraged by the systems according to the contexts. This phenomenon is similar in the second case. Translation given by NMT is not readable. In contrast, M-RefNet generates the core verb "strengthened" and B-RefNet provides a more accurate collocation "stepped up patrols".

4.3 Results on English-German Translation

On this task, DeepLAU (Wang et al., 2017b) is chosen as the baseline and also used as the pre-trained model. We list the translation performance of our models and some existing NMT systems in Table 5. All the systems except for Robust

NMT (Cheng et al., 2018) have a deep architecture with no less than 4 layers while Robust NMT introduces a additional discriminator for adversarial training. From the table, we can observe that our strong baseline DeepLAU is comparable to Google's neural machine translation (GNMT) (Wu et al., 2016). M-RefNet outperforms the baseline by 1.29 BLEU points and B-RefNet achieves slightly better performance with a 1.79 BLEU improvement, which is consistent to the results on Zh-En. Compared with the SOTA deep NMT systems, both M-RefNet and B-RefNet outperform GNMT and even obtain comparable performance with ConvS2S (Gehring et al., 2017) and Transformer (Vaswani et al., 2017) which have much deeper architectures with relative much more parameters. Since the reference networks do not rely on the recurrent structure, one interesting future direction is to apply our methods to such complicated models to bring further improvements.

5 Related Work

Document-level Neural Machine Translation

There are few works that consider the document-level contextual information to improve typical NMT. Jean et al. (2017) propose to use a additional encoder to generate the latent representation of previous sentence as extra context for decoder and attention mechanism is also applied between the decoder state and previous context to get access to word-level information of the previous sentence. Contemporaneously, Wang et al. (2017a) extend NMT by adding two encoders to encode the previous sentences in word-level and

	System	Architecture	BLEU
0	GNMT	8-layer LSTM encoder and decoder	24.60
1	Robust NMT	2-layer GRU encoder and decoder + adversarial training	25.26
2	ConvS2S	15-layer CNN encoder and decoder	25.16
3	Transformer (big)	6-layer encoder and decoder + 16-head self-attention	28.40
<i>This work</i>			
4	DeepLAU		24.37
5	M-RefNet	4-layer LAU encoder and decoder	25.66
6	B-RefNet		26.16

Table 5: Translation quality on En-De.

sentence-level respectively. The last hidden state of encoders are considered as the summarization of a previous sentence and the group. [Bawden et al. \(2018\)](#) employ multiple encoders to summarize the antecedent and propose to combine the contexts with a gated function. However, these incorporated extra encoders bring large amount of parameters and slow down the translation speed. [Tu et al. \(2018\)](#) propose to modify the NMT with light-weight key-value memory to store the translation history. However, due to the limitation of the memory size, the very short view on the previous (25 timesteps) is not sufficient to model the document-level contextual information. Additionally, [Maruf and Haffari \(2018\)](#) propose to capture the global source and target context of a entire document with memory network ([Graves et al., 2014](#); [Wang et al., 2016](#)). Nevertheless, since the number of sentence pairs in a document could be enormous, storing all sentence with memory components could be very time and space consuming. More recently, [Miculicich et al. \(2018\)](#) and [Zhang et al. \(2018\)](#) propose to improve Transformer by encoding previous sentences with extra encoders. The *reference book* in this work can be regarded as a special kind of document context. However, there are two major differences between our approach and the above work. First, we encode the entire corpus into a handful of anchor points which is much more light-weight but concentrated to capture the global contextual information. Second, the global contexts in this work is *static*. That means, given a sentence to translate, the final translation result only depends on the reference corpus, but not the sentences surrounding the current one.

Local Coding There are a number of works on manifold learning ([Roweis and Saul, 2000](#); [Van Gemert et al., 2008](#); [Yu et al., 2009](#); [Ladicky](#)

[and Torr, 2011](#)). The manifold learning methods approximate any point on the latent manifold with a linear combination of a set of localized anchor points relying on the assumption that high dimensional input usually lies on the lower dimensional manifold. [Agustsson et al. \(2017\)](#) utilize local coding into deep neural networks on age prediction from images and [Cao et al. \(2018\)](#) exploit LCC for GAN ([Goodfellow et al., 2014](#)) to capture the local information of data. All these works focus on application of Computer Vision while we apply LCC in a Nature Language Processing task. To our knowledge, this is the first attempt to incorporate local coding into NMT modeling.

6 Conclusion and Future Work

In this work, we propose two models to improve the translation quality of NMT inspired by the common human behaviors, paraphrasing and consulting. The monolingual model simulates the paraphrasing process by utilizing the global source information while the bilingual model provides a referable target word based on other sentence pairs in the training corpus. We conduct experiments on Chinese-English and English-German tasks, and the experimental results manifest the effectiveness and efficiency of our methods.

In the future, we would like to investigate the feasibility of our methods on non-recurrent NMT models such as Transformer ([Vaswani et al., 2017](#)). Moreover, we are also interested in incorporating discourse-level relations into our models.

Acknowledgments

We would like to thank the reviewers for their valuable comments and suggestions.

References

- Eirikur Agustsson, Radu Timofte, and Luc Van Gool. 2017. Anchored regression networks applied to age estimation and super resolution. In *Proceedings of ICCV*.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.
- Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of NAACL*.
- Jiezhong Cao, Yong Guo, Qingyao Wu, Chunhua Shen, and Mingkui Tan. 2018. Adversarial learning with local coordinate coding.
- Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766. Association for Computational Linguistics.
- Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation.
- Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of ICML*.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.
- Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.
- Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of EMNLP*.
- Lubor Ladicky and Philip Torr. 2011. Locally linear support vector machines. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 985–992.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*.
- Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of ACL*.
- Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954.
- Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326.
- 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 (Volume 1: Long Papers)*, volume 1, pages 1715–1725.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*.
- Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *Transactions of the Association of Computational Linguistics*, 6:407–420.
- Jan C Van Gemert, Jan-Mark Geusebroek, Cor J Veenman, and Arnold WM Smeulders. 2008. Kernel codebooks for scene categorization. In *Proceedings of ECCV*.
- 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 *Proceedings of NIPS*.
- Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017a. Exploiting cross-sentence context for neural machine translation. In *Proceedings of EMNLP*.
- Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2016. Memory-enhanced decoder for neural machine translation. In *Proceedings of EMNLP*.
- Mingxuan Wang, Zhengdong Lu, Jie Zhou, and Qun Liu. 2017b. Deep neural machine translation with linear associative unit. In *Proceedings of ACL*.
- 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*.

Kai Yu, Tong Zhang, and Yihong Gong. 2009. Nonlinear learning using local coordinate coding. In *Proceedings of NIPS*.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542.