Synchronous Bidirectional Neural Machine Translation

Long Zhou\textsuperscript{1,2}, Jiajun Zhang\textsuperscript{1,2,}\textsuperscript{*}, Chengqing Zong\textsuperscript{1,2,3}

\textsuperscript{1}National Laboratory of Pattern Recognition, CASIA, Beijing, China
\textsuperscript{2}University of Chinese Academy of Sciences, Beijing, China
\textsuperscript{3}CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai, China
{long.zhou, jjzhang, cqzong}@nlpr.ia.ac.cn

Abstract

Existing approaches to neural machine translation (NMT) generate the target language sequence token-by-token from left to right. However, this kind of unidirectional decoding framework cannot make full use of the target-side future contexts which can be produced in a right-to-left decoding direction, and thus suffers from the issue of unbalanced outputs. In this paper, we introduce a synchronous bidirectional–neural machine translation (SB-NMT) that predicts its outputs using left-to-right and right-to-left decoding simultaneously and interactively, in order to leverage both of the history and future information at the same time. Specifically, we first propose a new algorithm that enables synchronous bidirectional decoding in a single model. Then, we present an interactive decoding model in which left-to-right (right-to-left) generation does not only depend on its previously generated outputs, but also relies on future contexts predicted by right-to-left (left-to-right) decoding. We extensively evaluate the proposed SB-NMT model on large-scale NIST Chinese-English, WMT14 English-German, and WMT18 Russian-English translation tasks. Experimental results demonstrate that our model achieves significant improvements over the strong Transformer model by 3.92, 1.49, and 1.04 BLEU points, respectively, and obtains the state-of-the-art performance on Chinese-English and English-German translation tasks.\textsuperscript{1}

1 Introduction

Neural machine translation has significantly improved the quality of machine translation in recent years (Sutskever et al., 2014; Bahdanau et al., 2015; Zhang and Zong, 2015; Wu et al., 2016; Gehring et al., 2017; Vaswani et al., 2017). Recent approaches to sequence-to-sequence learning typically leverage recurrence (Sutskever et al., 2014), convolution (Gehring et al., 2017), or attention (Vaswani et al., 2017) as basic building blocks. Typically, NMT adopts the encoder-decoder architecture and generates the target translation from left to right. Despite their remarkable success, NMT models suffer from several weaknesses (Koehn and Knowles, 2017). One of the most prominent issues is the problem of unbalanced outputs in which the translation prefixes are better predicted than the suffixes (Liu et al., 2016). We analyze translation accuracy of the first and last 4 tokens for left-to-right (L2R) and right-to-left (R2L) directions, respectively. As shown in Table 1, the statistical results show that L2R performs better in the first 4 tokens, whereas R2L translates better in terms of the last 4 tokens. This problem is mainly caused by the left-to-right unidirectional decoding, which conditions each output word on previously generated outputs only, but leaving the future information from target-side contexts unexploited during translation. The future context is commonly used in reading and writing in human cognitive process

\textsuperscript{*}Corresponding author.

\textsuperscript{1}The source code is available at https://github.com/wszlong/sb-nmt.

91

Submission batch: 8/2018; Revision batch: 10/2018; Published 4/2019.
© 2019 Association for Computational Linguistics. Distributed under a CC-BY 4.0 license.
Table 1: Translation accuracy of the first 4 tokens and last 4 tokens in NIST Chinese-English translation tasks. L2R denotes left-to-right decoding and R2L means right-to-left decoding for conventional NMT.

<table>
<thead>
<tr>
<th>Model</th>
<th>The first 4 tokens</th>
<th>The last 4 tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>L2R</td>
<td>40.21%</td>
<td>35.10%</td>
</tr>
<tr>
<td>R2L</td>
<td>35.67%</td>
<td>39.47%</td>
</tr>
</tbody>
</table>

(Xia et al., 2017), and it is crucial to avoid under-translation (Tu et al., 2016; Mi et al., 2016).

To alleviate the problems, existing studies usually used independent bidirectional decoders for NMT (Liu et al., 2016; Sennrich et al., 2016a). Most of them trained two NMT models with left-to-right and right-to-left directions, respectively. Then, they translated and re-ranked candidate translations using two decoding scores together. More recently, Zhang et al. (2018) presented an asynchronous bidirectional decoding algorithm for NMT, which extended the conventional encoder-decoder framework by utilizing a backward decoder. However, these methods are more complicated than the conventional NMT framework because they require two NMT models or decoders. Furthermore, the L2R and R2L decoders are independent from each other (Liu et al., 2016), or only the forward decoder can utilize information from the backward decoder (Zhang et al., 2018). It is therefore a promising direction to design a synchronous bidirectional decoding algorithm in which L2R and R2L generations can interact with each other.

Accordingly, we propose in this paper a novel framework (SB-NMT) that utilizes a single decoder to bidirectionally generate target sentences simultaneously and interactively. As shown in Figure 1, two special labels (⟨l2r⟩ and ⟨r2l⟩) at the beginning of the target sentence guide translating from left to right or right to left, and the decoder in each direction can utilize the previously generated symbols of bidirectional decoding when generating the next token. Taking L2R decoding as an example, at each moment, the generation of the target word (e.g., \(y_3\)) does not only rely on previously generated outputs (\(y_1\) and \(y_2\)) of L2R decoding, but also depends on previously predicted tokens (\(y_n\) and \(y_{n-1}\)) of R2L decoding. Compared with the previous related NMT models, our method has the following advantages:

1) We use a single model (one encoder and one decoder) to achieve the decoding with left-to-right and right-to-left generation, which can be processed in parallel.
2) Via the synchronous bidirectional attention model (SBAtt, §3.2), our proposed model is an end-to-end joint framework and can optimize bidirectional decoding simultaneously.
3) Compared with two-phase decoding scheme in previous work, our decoder is faster and more compact, using one beam search algorithm.

Specifically, we make the following contributions in this paper:

- We propose a synchronous bidirectional NMT model that adopts one decoder to generate outputs with left-to-right and right-to-left directions simultaneously and interactively. To the best of our knowledge, this is the first work to investigate the effectiveness of a single NMT model with synchronous bidirectional decoding.

- Extensive experiments on NIST Chinese-English, WMT14 English-German and WMT18 Russian-English translation tasks demonstrate that our SB-NMT model obtains significant improvements over the strong Transformer model by 3.92, 1.49, and 1.04 BLEU points, respectively. In particular, our approach separately establishes the state-of-the-art BLEU score of 51.11 and 29.21 on Chinese-English and English-German translation tasks.

2 Background

In this paper, we build our model based on the powerful Transformer (Vaswani et al., 2017) with...
an encoder-decoder framework, where the encoder network first transforms an input sequence of symbols $x = (x_1, x_2, ..., x_n)$ to a sequence of continuous representations $z = (z_1, z_2, ..., z_n)$, from which the decoder generates an output sequence $y = (y_1, y_2, ..., y_m)$ one element at a time. Particularly, relying entirely on the multi-head attention mechanism, the Transformer with beam search algorithm achieves the state-of-the-art results for machine translation.

**Multi-head attention** allows the model to jointly attend to information from different representation subspaces at different positions. It operates on queries $Q$, keys $K$, and values $V$. For multi-head intra-attention of encoder or decoder, all of $Q, K, V$ are the output hidden-state matrices of the previous layer. For multi-head inter-attention of the decoder, $Q$ are the hidden states of the previous decoder layer, and $K$-$V$ pairs come from the output $(z_1, z_2, ..., z_n)$ of the encoder.

Formally, multi-head attention first obtains $h$ different representations of $(Q_i, K_i, V_i)$. Specifically, for each attention head $i$, we project the hidden-state matrix into distinct query, key, and value representations $Q_i = QW_i^Q$, $K_i = KW_i^K$, $V_i = VW_i^V$, respectively. Then we perform scaled dot-product attention for each representation, concatenate the results, and project the concatenation with a feed-forward layer.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_i)W^O$$

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

where $W_i^Q$, $W_i^K$, $W_i^V$, and $W^O$ are parameter projection matrices.

**Scaled dot-product attention** can be described as mapping a query and a set of key-value pairs to an output. Specifically, we can then multiply query $Q_i$ by key $K_i$ to obtain an attention weight matrix, which is then multiplied by value $V_i$ for each token to obtain the self-attention token representation. As shown in Figure 2, scaled dot-product attention operates on a query $Q$, a key $K$, and a value $V$ as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where $d_k$ is the dimension of the key. For the sake of brevity, we refer the reader to Vaswani et al. (2017) for more details.

**Standard Beam Search** Given the trained model and input sentence $x$, we usually employ beam search or greedy search (beam size = 1) to find the best translation $\hat{y} = \arg\max_y P(y|x)$. Beam size $N$ is used to control the search space by extending only the top-$N$ hypotheses in the current stack. As shown in Figure 3, the blocks represent the four best token expansions of the previous states, and these token expansions are sorted top-to-bottom from most probable to least probable. We define a complete hypothesis as a hypothesis which outputs EOS, where EOS is a special target token indicating the end of sentence. With the above settings, the translation $y$ is generated token-by-token from left to right.

### 3 Our Approach

In this section, we will introduce the approach of synchronous bidirectional NMT. Our goal is to
design a synchronous bidirectional beam search algorithm (§3.1) which generates tokens with both L2R and R2L decoding simultaneously and interactively using a single model. The central module is the synchronous bidirectional attention (SBAtt, see §3.2). By using SBAtt, the two decoding directions in one beam search process can help and interact with each other, and can make full use of the target-side history and future information during translation. Then, we apply our proposed SBAtt to replace the multi-head intra-attention in the decoder part of Transformer model (§3.3), and the model is trained end-to-end by maximum likelihood using stochastic gradient descent (§3.4).

3.1 Synchronous Bidirectional Beam Search

Figure 4 illustrates the synchronous bidirectional beam search process with beam size 4. With two special start tokens which are optimized during the training process, we let half of the beam keep decoding from left to right guided by the label \(\langle l_{2r}\rangle\), and allow the other half beam to decode from right to left, indicated by the label \(\langle r_{2l}\rangle\). More importantly, via the proposed SBAtt (§3.2) model, L2R (R2L) generation does not only depend on its previously generated outputs, but also relies on future contexts predicted by R2L (L2R) decoding.

Note that (1) at each time step, we choose the best items of the half beam from L2R decoding and the best items of the half beam from R2L decoding to continue expanding simultaneously; (2) L2R and R2L beams should be thought of as parallel, with SBAtt computed between items of 1-best L2R and R2L, items of 2-best L2R and R2L, and so on\(^2\); (3) the black blocks denote the ongoing expansion of the hypotheses, and decoding terminates when the end-of-sentence flag EOS is predicted; (4) in our decoding algorithm, the complete hypotheses will not participate in subsequent SBAtt, and the L2R hypothesis attended by R2L decoding may change at different time steps, while the ongoing partial hypotheses in both directions of SBAtt always share the same length; (5) finally, we output the translation result with highest probability from all complete hypotheses. Intuitively, our model is able to choose from L2R output or R2L output as final hypothesis according to their model probabilities, and if an R2L hypothesis wins, we reverse the tokens before presenting it.

3.2 Synchronous Bidirectional Attention

Instead of multi-head intra-attention which prevents future information flow in the decoder to preserve the auto-regressive property, we propose a synchronous bidirectional attention (SBAtt) mechanism. With the two key modules of synchronous bidirectional dot-product attention (§3.2.1) and synchronous bidirectional multi-head attention (§3.2.2), SBAtt is capable of capturing and combining the information generated by L2R and R2L decoding.

3.2.1 Synchronous Bidirectional Dot-Product Attention

Figure 5 shows our particular attention Synchronous Bidirectional Dot-Product Attention (SBDPA). The input consists of queries (\([Q^\rightarrow; Q^\leftarrow]\)), keys (\([K^\rightarrow; K^\leftarrow]\)), and values (\([V^\rightarrow; V^\leftarrow]\)) which are all concatenated by forward (L2R) states and backward (R2L) states. The new forward state \(\hat{H}\) and backward state \(\hat{H}\) can be obtained by

\(^2\) We also did experiments in which all of L2R hypotheses attend to the 1-best R2L hypothesis, and all the R2L hypotheses attend to the 1-best L2R hypothesis. The results of the two schemes are similar. For the sake of simplicity, we employed the previous scheme.
synchronous bidirectional dot-product attention. For the new forward state $\widehat{H}$, it can be calculated as:

$$
\widehat{H}_{\text{history}} = \text{Attention}(\widehat{Q}, \widehat{K}, \widehat{V})
$$

$\widehat{H}_{\text{future}} = \text{Attention}(\widehat{Q}, \widehat{K}, \widehat{V})
$$

$\widehat{H} = \text{Fusion}(\widehat{H}_{\text{history}}, \widehat{H}_{\text{future}})$

where $\widehat{H}_{\text{history}}$ is obtained by using conventional scaled dot-product attention as introduced in Equation 2, and its purpose is to take advantage of previously generated tokens, namely **history information**. We calculate $\widehat{H}_{\text{future}}$ using forward query ($\widehat{Q}$) and backward key-value pairs ($\widehat{K}, \widehat{V}$), which attempts at making use of **future information** from R2L decoding as effectively as possible in order to help predict the current token in L2R decoding. The role of Fusion($\cdot$) (green block in Figure 5) is to combine $\widehat{H}_{\text{history}}$ and $\widehat{H}_{\text{future}}$ by using linear interpolation, nonlinear interpolation, or gate mechanism.

**Linear Interpolation** $\widehat{H}_{\text{history}}$ and $\widehat{H}_{\text{future}}$ have different importance to prediction of current word. Linear interpolation of $\widehat{H}_{\text{history}}$ and $\widehat{H}_{\text{future}}$ produces an overall hidden state:

$$
\widehat{H} = \overline{\lambda} \widehat{H}_{\text{history}} + \lambda \widehat{H}_{\text{future}}
$$

where $\lambda$ is a hyper-parameter decided by the performance on development set.$^3$

**Nonlinear Interpolation** $\overline{\lambda}$ is equal to $\lambda_{\text{history}}$ in the conventional attention mechanism, and $\lambda_{\text{future}}$ means the attention information between current hidden state and generated hidden states of the other decoding. In order to distinguish two different information sources, we present a nonlinear interpolation by adding an activation function to the backward hidden states:

$$
\widehat{H} = \overline{\lambda}_{\text{history}} + \lambda * AF(\overline{\lambda}_{\text{future}})
$$

where $AF$ denotes activation function, such as $\tanh$ or $relu$.

**Gate Mechanism** We also propose a gate mechanism to dynamically control the amount of information flow from the forward and backward contexts. Specifically, we apply a feed-forward gating layer upon $\overline{\lambda}_{\text{history}}$ as well as $\overline{\lambda}_{\text{future}}$ to enrich the nonlinear expressiveness of our model:

$$
r_t, z_t = \sigma(\mathbf{W}_g(\overline{\lambda}_{\text{history}}, \overline{\lambda}_{\text{future}}))
$$

$$
\widehat{H} = r_t \odot \overline{\lambda}_{\text{history}} + z_t \odot \overline{\lambda}_{\text{future}}
$$

where $\odot$ denotes element-wise multiplication. Via this gating layer, it is able to control how much past information can be preserved from previous context and how much reversed information can be captured from backward hidden states.

Similar to the calculation of forward hidden states $\overline{\lambda}_i$, the backward hidden states $\overline{\lambda}_i$ can be computed as follows.

$$
\overline{\lambda}_{\text{history}} = \text{Attention}([\overline{\lambda}_i; \overline{\lambda}_i], [\overline{\lambda}_i; \overline{\lambda}_i])
$$

$$
\overline{\lambda}_{\text{future}} = \text{Attention}([\overline{\lambda}_i; \overline{\lambda}_i], [\overline{\lambda}_i; \overline{\lambda}_i])
$$

$$
\overline{\lambda} = \text{Fusion}(\overline{\lambda}_{\text{history}}, \overline{\lambda}_{\text{future}})
$$

where Fusion($\cdot$) is the same as introduced in Equations 4–6. Note that $\overline{\lambda}$ and $\overline{\lambda}$ can be calculated in parallel. We refer to the whole procedure formulated in Equation 3 and Equation 7 as SBDPA($\cdot$).

$$\overline{\lambda} = \text{SBDPA}([\overline{\lambda}_i; \overline{\lambda}_i], [\overline{\lambda}_i; \overline{\lambda}_i], [\overline{\lambda}_i; \overline{\lambda}_i])$$

$^3$Note that we can also set $\lambda$ to be a vector and learn $\lambda$ during training with standard back-propagation, and we remain it as future exploration.
3.2.2 Synchronous Bidirectional Multi-Head Attention

Multi-head attention consists of \( h \) attention heads, each of which learns a distinct attention function to attend to all of the tokens in the sequence, where mask is used for preventing leftward information flow in decoder. Compared with the multi-head attention, our inputs are the concatenation of forward and backward hidden states. We extend standard multi-headed attention by letting each head attend to both forward and backward hidden states, combined via SBDPA:

\[
\text{MultiHead}(\lbrack \vec{Q}; \vec{Q} \rbrack, \lbrack \vec{K}; \vec{K} \rbrack, \lbrack \vec{V}; \vec{V} \rbrack) = \text{Concat}(\lbrack \vec{H}_1; \vec{H}_1 \rbrack, \ldots, \lbrack \vec{H}_h; \vec{H}_h \rbrack) W^O \tag{9}
\]

and \( \lbrack \vec{H}_1; \vec{H}_1 \rbrack \) can be computed as follows, which is the biggest difference from conventional multi-head attention:

\[
\lbrack \vec{H}_i; \vec{H}_i \rbrack = \text{SBDPA}(\lbrack \vec{Q}; \vec{Q} \rbrack W^Q_i, \lbrack \vec{K}; \vec{K} \rbrack W^K_i, \lbrack \vec{V}; \vec{V} \rbrack W^V_i) \tag{10}
\]

where \( W^Q_i, W^K_i, W^V_i \) and \( W^O \) are parameter projection matrices, which are the same as standard multi-head attention introduced in Equation 1.

3.3 Integrating Synchronous Bidirectional Attention into NMT

We apply our synchronous bidirectional attention to replace the multi-head intra-attention in the decoder, as illustrated in Figure 6. The neural encoder of our model is identical to that of the standard Transformer model. From the source tokens, learned embeddings are generated which are then modified by an additive positional encoding. The encoded word embeddings are then used as input to the encoder which consists of \( N \) blocks each containing two layers: (1) a multi-head attention layer (MHAtt), and (2) a position-wise feed-forward layer (FFN).

The bidirectional decoder of our model is extended from the standard Transformer decoder. For each layer in the bidirectional decoder, the lowest sub-layer is our proposed synchronous bidirectional attention network, and it also uses residual connections around each of the sublayers, followed by layer normalization:

\[
s^l_d = \text{LayerNorm}(s^{l-1} + \text{SBAAtt}(s^{l-1}, s^{l-1}, s^{l-1})) \tag{11}
\]

and \( s^l_e = \text{LayerNorm}(s^l_e + \text{MHAtt}(s^l_e, h^N, h^N)) \)

\[
s^l = \text{LayerNorm}(s^l_e + \text{FFN}(s^l_e)) \tag{12}
\]

where \( l \) denotes layer depth, and subscript \( d \) means the decoder-informed intra-attention representation; SBAAtt is our proposed synchronous bidirectional attention, and \( s^{l-1} \) is equal to \( \lbrack \vec{s}^{l-1}; \vec{s}^{l-1} \rbrack \) containing forward and backward hidden states. In addition, the decoder stacks another two sub-layers to seek translation-relevant source semantics to bridge the gap between the source and target language:

\[
s^l_e = \text{LayerNorm}(s^l_d + \text{MHAtt}(s^l_d, h^N, h^N))
\]

where MHAtt denotes the multi-head attention introduced in Equation 1, and we use \( e \) to denote the encoder-informed inter-attention representation; \( h^N \) is the source top layer hidden state, and FFN means feed-forward networks.

Finally, we use a linear transformation and softmax activation to compute the probability of the next tokens based on \( s^N = \lbrack \vec{s}^N; \vec{s}^N \rbrack \),
namely the final hidden states of forward and backward decoding:
\[
\begin{align*}
 p(\vec{y}_j | \vec{y}_{<j}, \vec{y}_{<j}, x, \theta) &= \text{Softmax}(\vec{s}^N_W) \\
 p(\vec{y}_j | \vec{y}_{<j}, \vec{y}_{<j}, x, \theta) &= \text{Softmax}(\vec{s}^W_N)
\end{align*}
\]
where \(\theta\) is shared weight for L2R and R2L decoding, and \(W\) is the weight matrix.

3.4 Training

We design a simple yet effective strategy to enable synchronous bidirectional translation within a decoder. We separately add the special labels \((\langle \mathrm{L2R}\rangle\) and \((\langle \mathrm{R2L}\rangle)\) at the beginning of target sentence \((\vec{y}_1)\) and \((\vec{y}_2)\) to guide translating from left to right or right to left. Given a set of training examples \(\{x^{(z)}, y^{(z)}\}_{z=1}^{Z}\), the training algorithm aims to find the model parameters that maximize the likelihood of the training data:
\[
J(\theta) = \frac{1}{Z} \sum_{z=1}^{Z} \sum_{j=1}^{M} \left\{ \log p(\vec{y}_j^{(z)} | \vec{y}_{<j}^{(z)}, y_{<j}^{(z)}, x^{(z)}, \theta) + \log p(\vec{y}_j^{(z)} | \vec{y}_{<j}^{(z)}, \vec{y}_{<j}^{(z)}, x^{(z)}, \theta) \right\}
\]

Similar to asynchronous bidirectional decoding (Zhang et al., 2018) and bidirectional language models in BERT (Devlin et al., 2018), the proposed SB-NMT model also faces the same training problem: the bidirectional decoding would allow the words (the second half of the decoding sequence) to indirectly “see themselves” from the other decoding direction. To ensure consistency between model training and testing, we construct pseudo references \(\vec{y}_p (\vec{y}_p)\) for gold \(\vec{y}_g (\vec{y}_g)\). More specifically, we first train an L2R model using \((x, \vec{y}_g)\) and an R2L model using \((x, \vec{y}_g)\). Then we use the two models to translate source sentences \(x\) into pseudo target sentences \(\vec{y}_p\) and \(\vec{y}_p\), respectively. Finally, we get two triples \((x, \vec{y}_p, \vec{y}_g)\) and \((x, \vec{y}_g, \vec{y}_p)\) as our training data.

Once the proposed model is trained, we employ the bidirectional beam search algorithm to predict the target sequence, as illustrated in Figure 4. Compared with previous work that usually adopts a two-phase scheme to translate input sentences (Liu et al., 2016; Sennrich et al., 2017; Zhang et al., 2018), our decoding approach is more compact and effective.

4 Experiments

We evaluate the proposed model on three translation datasets with different sizes, including NIST Chinese-English, WMT14 English-German, and WMT18 Russian-English translations.

4.1 Datasets

For Chinese-English, our training data includes about 2.0 million sentence pairs extracted from the LDC corpus.\(^4\) We use the NIST 2002 (MT02) Chinese-English dataset as the validation set and NIST 2003-2006 (MT03-06) as our test set. We use BPE (Sennrich et al., 2016b) to encode Chinese and English, respectively. We learn 30K merge operations and limit the source and target vocabularies to the most frequent 30K tokens.

For English-German translation, the training set consists of about 4.5 million bilingual sentence pairs from WMT 2014.\(^5\) We use newstest2013 as the validation set and newstest2014 as the test set. Sentences are encoded using BPE, which has a shared vocabulary of about 37,000 tokens. To evaluate the models, we compute the BLEU metric (Papineni et al., 2002) on tokenized, true-case output.\(^6\)

For Russian-English translation, we use the following resources from the WMT parallel data:\(^7\): ParaCrawl corpus, Common Crawl corpus, News Commentary v13, and Yandex Corpus. We do not use Wiki Headlines and UN Parallel Corpus V1.0. The training corpus consists of 14M sentence pairs. We employ the Moses Tokenizer\(^8\) for preprocessing. For subword segmentation, we use 50,000 joint BPE operations and choose the most frequent 52,000 tokens as vocabularies. We use newstest2017 as the development set and the newstest2018 as the test set.

\(\text{The corpora includes LDC2000T50, LDC2002T01, LDC2002E18, LDC2003E07, LDC2003E14, LDC2003T17, and LDC2004T07. Following previous work, we also use case-insensitive tokenized BLEU to evaluate Chinese-English which have been segmented by Stanford word segmentation and Moses Tokenizer, respectively.}\)

\(\text{http://www.statmt.org/wmt14/translation-task.html. All preprocessed datasets and vocab can be directly download in tensor2tensor website https://drive.google.com/open?id=0B_bZck-ksdkpM25jRUN2X2UxMn8.}\)

\(\text{This procedure is used in the literature to which we compare (Wu et al., 2016; Gehring et al., 2017; Vaswani et al., 2017).}\)

\(\text{http://www.statmt.org/wmt18/translation-task.html.}\)

\(\text{https://github.com/moses-smt/mosesdecoder/blobv/master/scripts/tokenizer/tokenizer.perl.}\)
4.2 Setting
We build the described models by modifying the tensor2tensor\footnote{https://github.com/tensorflow/tensor2tensor.} toolkit for training and evaluating. For our bidirectional Transformer model, we employ the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.998$, and $\epsilon = 10^{-9}$. We use the same warmup and decay strategy for learning rate as Vaswani et al. (2017), with 16,000 warmup steps. During training, we employ label smoothing of value $\epsilon_{ls}=0.1$. For evaluation, we use beam search with a beam size of $k = 4$. For SB-NMT, we use two L2R and R2L hypotheses, respectively, and length penalty $\alpha = 0.6$. Additionally, we use 6 encoder and decoder layers, hidden size $d_{model} = 1,024$, 16 attention heads, 4,096 feedforward inner-layer dimensions, and $P_{dropout} = 0.1$. Our settings are close to transformer\_big setting as defined in Vaswani et al. (2017). We employ three Titan Xp GPUs to train English-German and Russian-English translation, and one GPU for Chinese-English translation pairs. In addition, we use a single model obtained by averaging the last 20 checkpoints for English-German and Russian-English and do not perform checkpoint averaging for Chinese-English.

4.3 Baselines
We compare the proposed model against the following state-of-the-art statistical machine translation (SMT) and NMT systems\footnote{For fair comparison, Rerank-NMT and ABD-NMT are based on strong Transformer models.}:

- **Moses**: an open source phrase-based SMT system with default configuration and a 4-gram language model trained on the target portion of training data.
- **RNMT** (Luong et al., 2015): it is a state-of-the-art RNN-based NMT system with default setting.
- **Transformer**: it has obtained the state-of-the-art performance on machine translation, which predicts target sentence from left to right relying on self-attention (Vaswani et al., 2017).
- **Transformer (R2L)**: it is a variant of Transformer that generates translation in a right-to-left direction.

<table>
<thead>
<tr>
<th>Fusion</th>
<th>$\lambda = 0.1$</th>
<th>$\lambda = 0.5$</th>
<th>$\lambda = 1.0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>51.05</td>
<td>50.71</td>
<td>46.98</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>$tanh$</td>
<td>50.99</td>
<td>50.72</td>
</tr>
<tr>
<td></td>
<td>$relu$</td>
<td>50.79</td>
<td>50.57</td>
</tr>
<tr>
<td>Gate</td>
<td></td>
<td>50.51</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Experiment results on the development set using different fusion mechanisms with different $\lambda$s.

- **Rerank-NMT**: Via exploring the agreement on left-to-right and right-to-left NMT models (Liu et al., 2016; Sennrich et al., 2016a), first run beam search for forward and reverse models independently to obtain two $k$-best lists, and then re-score the union of two $k$-best lists ($k = 10$ in our experiments) using the joint model (adding logprobs) to find the best candidate.

- **ABD-NMT**: it is an asynchronous bidirectional decoding for NMT, which equipped the conventional attentional encoder-decoder NMT model with a backward decoder (Zhang et al., 2018). ABD-NMT adopts a two-phrase decoding scheme: (1) use backward decoder to generate reverse sequence states; (2) perform beam search on the forward decoder to find the best translation based on encoder hidden states and backward sequence states.

4.4 Results on Chinese-English Translation

**Effect of Fusion Mechanism** We first investigate the impact of different fusion mechanisms with different $\lambda$s on the development set. As shown in Table 2, we find that linear interpolation is sensitive to parameters $\lambda$. Nonlinear interpolation, which is more robust than linear interpolation, achieves the best performance when we use $tanh$ with $\lambda = 0.1$. Compared with gate mechanism, nonlinear interpolation is much simpler and needs less parameters. Therefore, we will use nonlinear interpolation with $tanh$ and $\lambda = 0.1$ for all experiments thereafter.

**Translation Quality** Table 3 shows translation performance for Chinese-English. Specifically, the proposed model significantly outperforms Moses, RNMT, Transformer, Transformer (R2L), Rerank-NMT, and ABD-NMT by 13.23, 8.54, 3.92, 4.90, 2.91, and 2.82 BLEU points, respectively. Compared with Transformer and Transformer (R2L), our model exhibits much better
performance. These results confirm our hypothesis that the two directions are mutually beneficial in bidirectional decoding. Furthermore, compared with Rerank-NMT in which two decoders are relatively independent and ABD-NMT where only the forward decoder can rely on a backward decoder, our proposed model achieves substantial improvements over them on all test sets, which indicates that joint modeling and optimizing with left-to-right and right-to-left decoding behaves better in leveraging bidirectional decoding.

4.5 Results on English-German Translation

We further demonstrate the effectiveness of our model in WMT14 English-German translation tasks, and we also display the performances of some competitive models including GNMT (Wu et al., 2016), Conv (Gehring et al., 2017), and AttIsAll (Vaswani et al., 2017). As shown in Table 4, our model also significantly outperforms others and gets an improvement of 1.49 more

<table>
<thead>
<tr>
<th>Model</th>
<th>DEV</th>
<th>MT03</th>
<th>MT04</th>
<th>M05</th>
<th>MT06</th>
<th>AVE</th>
<th>∆</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moses</td>
<td>37.85</td>
<td>37.47</td>
<td>41.20</td>
<td>36.41</td>
<td>36.03</td>
<td>37.78</td>
<td>-9.41</td>
</tr>
<tr>
<td>RNMT</td>
<td>42.43</td>
<td>42.43</td>
<td>44.56</td>
<td>41.94</td>
<td>40.95</td>
<td>42.47</td>
<td>-4.72</td>
</tr>
<tr>
<td>Transformer</td>
<td>48.12</td>
<td>47.63</td>
<td>48.32</td>
<td>47.51</td>
<td>45.31</td>
<td>47.19</td>
<td>-</td>
</tr>
<tr>
<td>Transformer (R2L)</td>
<td>47.81</td>
<td>46.79</td>
<td>47.01</td>
<td>46.50</td>
<td>44.13</td>
<td>46.11</td>
<td>-1.08</td>
</tr>
<tr>
<td>Rerank-NMT</td>
<td>49.18</td>
<td>48.23</td>
<td>48.91</td>
<td>48.73</td>
<td>46.51</td>
<td>48.10</td>
<td>+0.91</td>
</tr>
<tr>
<td>ABD-NMT</td>
<td>48.28</td>
<td>49.47</td>
<td>48.01</td>
<td>48.19</td>
<td>47.09</td>
<td>48.19</td>
<td>+1.00</td>
</tr>
<tr>
<td>Our Model</td>
<td>50.99</td>
<td>51.87</td>
<td>51.50</td>
<td>51.23</td>
<td>49.83</td>
<td>51.11</td>
<td>+3.92</td>
</tr>
</tbody>
</table>

Table 3: Evaluation of translation quality for Chinese-English translation tasks using case-insensitive BLEU scores. All results of our model are significantly better than Transformer and Transformer (R2L) (p < 0.01).

<table>
<thead>
<tr>
<th>Model</th>
<th>TEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNMT‡ (Wu et al., 2016)</td>
<td>24.61</td>
</tr>
<tr>
<td>Conv‡ (Gehring et al., 2017)</td>
<td>25.16</td>
</tr>
<tr>
<td>AttIsAll‡ (Vaswani et al., 2017)</td>
<td>28.40</td>
</tr>
<tr>
<td>Transformer11</td>
<td>27.72</td>
</tr>
<tr>
<td>Transformer (R2L)</td>
<td>27.13</td>
</tr>
<tr>
<td>Rerank-NMT</td>
<td>27.81</td>
</tr>
<tr>
<td>ABD-NMT</td>
<td>28.22</td>
</tr>
<tr>
<td>Our Model</td>
<td>29.21</td>
</tr>
</tbody>
</table>

Table 4: Results of WMT14 English-German translation using case-sensitive BLEU. Results with ‡ mark are taken from the corresponding papers.

4.6 Results on Russian-English Translation

Table 5 shows the results of large-scale WMT18 Russian-English translation, and our approach still significantly outperforms the state-of-the-art Transformer model in development and test sets by 1.10 and 1.04 BLEU points, respectively. Note that the BLEU score gains of English-German and Russian-English are not as significant as that on Chinese-English. The underlying reasons, which have also been mentioned in Shen et al. (2016) and Zhang et al. (2018), are that (1) the Chinese-English datasets contain four reference translations for each source sentence while the English-German and Russian-English datasets only have a single reference; (2) English is more distantly related to Chinese than German and Russian, leading to the predominant improvements for Chinese-English translation when leveraging bidirectional decoding.

4.7 Analysis

We conduct analyses on Chinese-English translation to better understand our model from different perspectives.

Parameters and Speeds In contrast to the standard Transformer, our model does not increase any parameters except for a hyper-parameter $\lambda$, as

---

11 The BLEU scores for Transformer model are our reproduced results. Similar to footnote 7 in Chen et al. (2018), our performance is slightly lower than those reported in Vaswani et al. (2017). Additionally, we only use 3 GPUs for English-German, whereas most papers employ 8 GPUs for model training.
Table 6: Statistics of parameters, training, and testing speeds. \textit{Train} denotes the number of global training steps processed per second at the same batch-size sentences; \textit{Test} indicates the amount of translated sentences in 1 second.

![Table 6](image)

shown in Table 6. Rerank-NMT needs to train two sets of NMT models, so its parameters are doubled. The parameters of ABD-NMT are 333.8M since it has two decoders containing a backward decoder and a forward decoder. Hence, our model is more compact because it only has a single encoder-decoder NMT model.

We also show the training and testing speed of our model and baselines in Table 6. During training, our model performs approximately 1.26 training steps per second, which is faster than Rerank-NMT and ABD-NMT. When it comes to decoding procedure, the decoding speed of our model is 17.87 sentences per second with batch size 50, which is two or three times faster than Rerank-NMT and ABD-NMT.

**Effect of Unbalanced Outputs** According to Table 1, L2R usually does well on predicting the left-side tokens of target sequences, while R2L usually performs well on the right-side tokens. Our central idea is to combine the advantage of left-to-right and right-to-left modes. To test our hypothesis, we further analyze the translation accuracy of Rerank-NMT, ABD-NMT, and our model, as shown in Figure 7. Rerank-NMT and ABD-NMT can alleviate the unbalanced output problem, but fail to improve prefix and suffix accuracies at the same time. The experimental results demonstrate that our model can balance the outputs, and gets the best translation accuracy for both the first four words and the last four words. Note that our model chooses from L2R output or R2L output as final results according to their model probabilities, and the left-to-right decoding contributes 58.6% on test set.

**Effect of Varying Beam Size** We observe that beam search decoding only improves translation quality for narrow beams and degrades translation quality when exposed to a larger search space for L2R and R2L decoding, as illustrated in Figure 8. Additionally, the gap between greedy search and beam search is significant and can be up to about 1–2 BLEU points. Koehn and Knowles (2017) also demonstrate these phenomena in eight translation directions.

As for our SB-NMT model, we investigate the effect of different beam sizes \( k \), as shown by the red line of Figure 8. Compared with conventional beam search, where worse translations are found beyond an optimal beam size setting (e.g., in the range of 4–32), the translation quality of our proposed model remains stable as beam size becomes larger. We attribute this to the ability of

\[ \text{BLEU Score} \]

Figure 7: Translation accuracy of the first and last 4 tokens for Transformer, Transformer (R2L), Rerank-NMT, ABD-NMT, and our proposed model.

Figure 8: Translation qualities (BLEU score) of our L2R, R2L, and our SB-NMT model as beam size becomes larger\(^\text{12}\).

\[ \text{Effect of Varying Beam Size} \]

For greedy search in SB-NMT, it has one item L2R decoding and one item R2L decoding. In other words, its beam size is equal to 2 compared to conventional beam search decoding.
Figure 9: Performance of translations on the test set with respect to the lengths of the source sentences.

**Effect of Long Sentences** A well-known flaw of NMT models is the inability to properly translate long sentences. We follow Bahdanau et al. (2015) by grouping sentences of similar lengths together and compute a BLEU score per group (left picture). Figure 9 shows the BLEU score and the averaged length of translations for each group (right picture). Transformer and Transformer (R2L) perform very well on short source sentences, but degrade on long source sentences. Our model can alleviate this problem by taking advantage of both history and future information. In fact, incorporating synchronous bidirectional attention boosts translation performance on all source sentence groups.

**Subjective Evaluation** We follow Tu et al. (2016) by conducting a subjective evaluation to validate the benefit of the synchronous bidirectional decoder, as shown in Table 7. Four human evaluators are asked to evaluate the translations of 100 source sentences, which are randomly sampled from the test sets, without knowing which system the translation is selected from. These 100 source sentences have 2,712 words. We evaluate over- or under-translation based on the number of source words which are dropped or repeated in translation, though we use subword (Senrich et al., 2016b) in training and inference. Transformer and Transformer (R2L) suffer from serious under-translation problems with 7.85% and 7.81% errors. Our proposed model alleviates the under-translation problems by exploiting the combination of left-to-right and right-to-left decoding directions, reducing 30.6% of under-translation errors. It should be emphasized that the proposed model is especially effective for alleviating the under-translation problem, which is a more serious translation problem for Transformer systems, as seen in Table 7.

**Case Study** Table 8 gives three examples to show the translations of different models, in order to better understand how our model outperforms others. We find that Transformer produces translations with good prefixes (red line or dotted line), whereas Transformer (R2L) generates translations with better suffixes (blue line or wave line). Therefore, they are often unable to translate the whole sentence precisely. In contrast, the proposed approach can make full use of bidirectional decoding and remedy the errors in these cases.

5 Related Work

Our research is built upon a sequence-to-sequence model (Vaswani et al., 2017), but it is also related
to future modeling and bidirectional decoding. We discuss these topics in the following.

**Future Modeling** Standard neural sequence decoders generate target sentences from left to right, and it has been proven to be important to establish the direct information flow between currently predicted word and previously generated words (Zhou et al., 2017b; Vaswani et al., 2017). However, current methods still fail to estimate some desired information in the future. To address this problem, reinforcement learning methods have been applied to predict future properties (Li et al., 2017; Bahdanau et al., 2017; He et al., 2017). Li et al. (2018) presented a target foresight based attention which uses the POS tag as the partial information of a target foresight word to improve alignment and translation. Inspired by human cognitive behaviors, Xia et al. (2017) proposed a deliberation network, which leverages global information by observing both back and forward information in sequence decoding through a deliberation process. Zheng et al. (2018) introduced two additional recurrent layers to model translated past contents and untranslated future contents. The most relevant models in future modeling are twin networks (Serdyuk et al., 2018), which encourage the hidden state of the forward network to be close to that of the backward network used to predict the same token. However, they still used two decoders, and the backward network contributes nothing during inference. Along the direction of future modeling, we introduce a single synchronous bidirectional decoder, where forward decoding can be used as future information for backward decoding, and vice versa.

**Bidirectional Decoding** In SMT, many approaches explored backward language models or target-bidirectional decoding to capture right-to-left target-side contexts for translation (Watanabe and Sumita, 2002; Finch and Sumita, 2009; Zhang et al., 2013). To address the issue of unbalanced outputs, Liu et al. (2016) proposed an agreement model to encourage the agreement between L2R and R2L NMT models. Similarly, some work attempted to re-rank the left-to-right decoding results by right-to-left decoding, leading to diversified translation results (Sennrich et al., 2016a; Hoang et al., 2017; Tan et al., 2017; Sennrich et al., 2017; Liu et al., 2018; Deng et al., 2018). Recently, Zhang et al. (2018) proposed asynchronous bidirectional decoding for NMT, which extended the conventional attentional encoder-decoder framework by introducing a backward decoder. Additionally, both Niehues et al. (2016) and Zhou et al. (2017a) combined the strengths of NMT and SMT, which can also be used to combine the advantages of bidirectional translation texts (Zhang et al., 2018). Compared with previous methods, our method has the following advantages: (1) We use a single model to achieve the goal of synchronous left-to-right and right-to-left decoding. (2) Our model can leverage and combine the two decoding directions in every layer of the Transformer decoder, which can run in parallel. (3) By using synchronous bidirectional attention, our model is an end-to-end joint framework and can optimize L2R and R2L decoding simultaneously. (4) Compared with two-phase decoding schemes in previous work, our decoder is more compact and faster.
6 Conclusions and Future Work

In this paper, we propose a synchronous bidirectional NMT model that performs bidirectional decoding simultaneously and interactively. The bidirectional decoder, which can take full advantage of both history and future information provided by bidirectional decoding states, predicts its outputs by using left-to-right and right-to-left directions at the same time. To the best of our knowledge, this is the first attempt to integrate synchronous bidirectional attention into a single NMT model. Extensive experiments demonstrate the effectiveness of our proposed model. Particularly, our model respectively establishes state-of-the-art BLEU scores of 51.11 and 29.21 on NIST Chinese-English and WMT14 English-German translation tasks. In future work, we plan to apply this framework to other tasks, such as sequence labeling, abstractive summarization, and image captioning.

Acknowledgments

We thank the anonymous reviewers as well as the Action Editor, George Foster, for insightful comments and suggestions. The research work has been funded by the Natural Science Foundation of China under Grant No. 61673380. This work is also supported by grants from NVIDIA NVAIL program.

References


