

Sequence Generation: From Both Sides to the Middle

Long Zhou^{1,2}, Jiajun Zhang^{1,2}, Chengqing Zong^{1,2,3} and Heng Yu⁴

¹University of Chinese Academy of Sciences, Beijing, China

²National Laboratory of Pattern Recognition, CASIA, Beijing, China

³CAS Center for Excellence in Brain Science and Intelligence Technology, Shanghai, China

⁴Machine Intelligence Technology Lab, Alibaba Group

{long.zhou, jjzhang, cqzong}@nlpr.ia.ac.cn, yuheng.yh@alibaba-inc.com

Abstract

The encoder-decoder framework has achieved promising process for many sequence generation tasks, such as neural machine translation and text summarization. Such a framework usually generates a sequence token by token from left to right, hence (1) this autoregressive decoding procedure is time-consuming when the output sentence becomes longer, and (2) it lacks the guidance of future context which is crucial to avoid under-translation. To alleviate these issues, we propose a synchronous bidirectional sequence generation (SBSG) model which predicts its outputs from both sides to the middle simultaneously. In the SBSG model, we enable the left-to-right (L2R) and right-to-left (R2L) generation to help and interact with each other by leveraging interactive bidirectional attention network. Experiments on neural machine translation (En⇒De, Ch⇒En, and En⇒Ro) and text summarization tasks show that the proposed model significantly speeds up decoding while improving the generation quality compared to the autoregressive Transformer.

1 Introduction

The neural encoder-decoder framework has been widely adopted in sequence generation tasks, including neural machine translation (NMT) [Sutskever *et al.*, 2014; Bahdanau *et al.*, 2015; Vaswani *et al.*, 2017], text summarization [Rush *et al.*, 2015; Zhou *et al.*, 2017; Li *et al.*, 2018], and image captioning [Xu *et al.*, 2015; Vinyals *et al.*, 2015]. In this framework, the encoder models the semantics of the input sentence and transforms it into a context vector representation, which is then fed into the decoder to generate the output sequence token by token in a left-to-right manner.

Although the framework has obtained great success, the sequence-to-sequence model suffers from the decoding efficiency problem [Gu *et al.*, 2017]. Most of the models use autoregressive decoders that operate one step at a time, and they become slow when generating long sequences because a computationally intensive neural network is used to predict each token. Several recently proposed models avoid recurrence at training time by leveraging convolutions [Gehring *et*

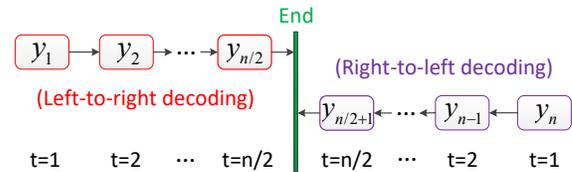


Figure 1: The illustration of synchronous bidirectional decoding for sequence-to-sequence model. The bidirectional decoder, predicting its outputs from left to right and from right to left simultaneously and interactively, can produce two tokens at each time step.

al., 2017] or self-attention [Vaswani *et al.*, 2017] as more-parallelizable alternatives to recurrent neural networks, but the decoding process cannot share the speed strength of parallelization due to the autoregressive generation schema in the decoder. More importantly, this left-to-right decoding cannot take advantage of future contexts which can be generated in a right-to-left decoding [Zhang *et al.*, 2018b].

To avoid this autoregressive property, Gu *et al.* [2017] proposed a non-autoregressive model to speed up machine translation by directly generating target words without relying on any previous predictions. Oord *et al.* [2017] modified a convolutional network for non-autoregressive modeling of speech synthesis. Lee *et al.* [2018] introduced a conditional non-autoregressive neural sequence model based on iterative refinement. However, in spite of their improvement in decoding speed, non-autoregressive models typically suffer from the substantial drop in generation quality.

In this paper, we propose a synchronous bidirectional sequence generation (SBSG) model to achieve a better improvement on both generation quality and decoding speed. Instead of producing output sentences token by token or predicting its outputs in a totally parallel manner, the SBSG model generates two tokens at a time. As shown in Figure 1, the bidirectional decoder can generate output sentences from both sides to the middle with both left-to-right (L2R) and right-to-left (R2L) directions. Furthermore, we introduce an interactive bidirectional attention network to bridge L2R and R2L outputs. More specifically, at each moment, the generation of target-side tokens does not only rely on its previously generated outputs (history information), but also depends on previously predicted tokens of the other generation direction (future information).

Specifically, the contributions of this paper can be summa-

rized as two folds:

- We propose a novel SBSG model that employs one decoder to predict outputs from both sides to the middle simultaneously and interactively. To the best of our knowledge, this is the first work to perform sequence generation from both ends to the middle.
- We extensively evaluate the proposed model on typical sequence generation tasks, namely neural machine translation and text summarization. In the case of **machine translation**, we not only obtain approximately $1.4\times$ ($1.5\times$) speedup for decoding than autoregressive Transformer with beam search (greedy search), but also get an improvement of 0.39 (0.99), 1.26 (2.87) and 0.73 (1.11) BLEU points of translation quality in WMT14 En \Rightarrow De, NIST Ch \Rightarrow En and WMT16 En \Rightarrow Ro respectively, which also significantly outperforms previous non-autoregressive models [Gu *et al.*, 2017; Lee *et al.*, 2018; Kaiser *et al.*, 2018]. For **text summarization**, the proposed model is able to decode approximately $1.5\times$ faster while achieving better generation quality relative to the autoregressive counterparts.

2 Related Work

Autoregressive Decoding. Recent approaches to sequence to sequence learning typically leverage recurrence [Sutskever *et al.*, 2014], convolution [Gehring *et al.*, 2017], or self-attention [Vaswani *et al.*, 2017] as basic building blocks. Particularly, relying entirely on the attention mechanism, the Transformer introduced by Vaswani *et al.* [2017] can improve the training speed as well as model performance. To accelerate autoregressive architecture, Mi *et al.* [2016] introduced a sentence-level vocabulary which is able to reduce computing time and memory usage. Devlin [2017] focused on fast and accurate neural machine translation decoding in CPU. Zhang *et al.* [2018a] proposed an average attention network (AAN) as an alternative to the self-attention network in the decoder of Transformer. Despite their remarkable success, they are difficult to parallelize and this unidirectional decoding framework limits its potential [Liu *et al.*, 2016a].

Non-Autoregressive Decoding. In terms of speeding up the decoding of the neural Transformer, Gu *et al.* [2017] modified the autoregressive architecture to speed up machine translation by directly generating target words in parallel. However, the main drawback of this work is the need for extensive policy gradient fine-tuning techniques, as well as the issue that this method only works for machine translation and cannot be applied to other sequence generation tasks. In parallel to Gu *et al.* [2017], Oord *et al.* [2017] presented a successful, non-autoregressive sequence model for speech waveform. Besides, Kaiser *et al.* [2018] first auto-encoded the target sequence into a shorter sequence of discrete latent variables, and then decoded the output sentence from this shorter latent sequence in parallel. Lee *et al.* [2018] introduced a conditional non-autoregressive neural sequence model based on iterative refinement. Concurrently to our work, Wang *et al.* [2018] presented a semi-autoregressive Transformer for faster translation without changing the autoregressive proper-

ty in global. However, these approaches improved the parallelizability but significantly reduced generation quality.

Towards Bidirectional Decoding. Liu *et al.* [2016a] proposed an agreement model to encourage the agreement between a pair of target-directional LSTMs, which generated more balanced targets. Similarly, some work attempted at target-bidirectional decoding for SMT or NMT [Watanabe and Sumita, 2002; Finch and Sumita, 2009; Liu *et al.*, 2016b; Sennrich *et al.*, 2016a; Liu *et al.*, 2018]. Recently, Zhang *et al.* [2018b] and Zhou *et al.* [2019] proposed an asynchronous and synchronous bidirectional decoding for NMT, respectively. Serdyuk *et al.* [2018] presented the twin networks to encourage the hidden state of the forward network to be close to that of the backward network used to predict the same token. Nevertheless, the above studies are not to speed up the decoding procedure, and even sacrifice speed in exchange for quality improvement. Our work differs from those by introducing a novel sequence generation model which aims at taking full advantage of both left-to-right and right-to-left decoding to accelerate and improve sequence generation.

3 The Framework

Our goal in this work is to achieve a better improvement on both generation quality and decoding speed. We introduce a novel method for decoding with both left-to-right and right-to-left manners simultaneously and interactively in a unified model. As demonstrated in Figure 2, our proposed model consists of an encoder and a bidirectional decoder, in which two special labels ($\langle l2r \rangle$ and $\langle r2l \rangle$) at the beginning of output sentence are utilized to guide the sequence generation from left to right or right to left. The bidirectional decoder reads the encoder representation and generates two output tokens at each time step, by using interactive bidirectional attention networks. Next, we will detail individual components and introduce an algorithm for training and inference.

3.1 The Neural Encoder

Given an input sentence $x = (x_1, x_2, \dots, x_m)$, the new Transformer leverages its encoder to induce input-side semantic and dependencies so as to enable its decoder to recover the encoded information in an output sentence. The encoder is composed of a stack of N identical layers, each of which has two sub-layers:

$$\begin{aligned} \tilde{h}^l &= \text{LN}(h^{l-1} + \text{MHAtt}(h^{l-1}, h^{l-1}, h^{l-1})) \\ h^l &= \text{LN}(\tilde{h}^l + \text{FFN}(\tilde{h}^l)) \end{aligned} \tag{1}$$

where the superscript l indicates layer depth, LN is layer normalization, FFN means feed-forward networks, and MHAtt denotes the multi-head attention mechanism as follows.

Scaled Dot-Product Attention. An attention function can be described as mapping a query and a set of key-value pairs to an output. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query and the corresponding key. Scaled dot-product attention operates on a query Q , key K , and a value V as:

$$\text{ATT}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

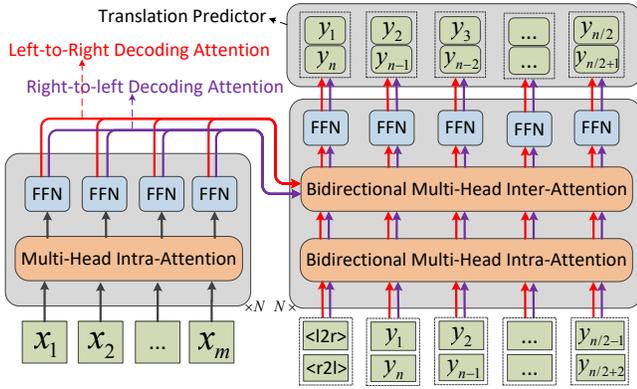


Figure 2: The new Transformer architecture with the proposed bidirectional multi-head (intra- and inter-) attention network. Instead of producing output sentence token by token or predicting its outputs in totally parallel, the proposed model generates two tokens (one from left to right, the other one from right to left) at a time, indicated by two special labels.

where d_k is the dimension of the key.

Multi-Head Attention. We use the multi-head version with h heads. It obtains h different representations of (Q, K, V) , computes scaled dot-product attention for each representation, concatenates the results, and projects the concatenation with a feed-forward layer.

$$\text{MHAtt}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

$$\text{head}_i = \text{ATT}(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

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

3.2 The Bidirectional Decoder

The bidirectional decoder performs decoding in both left-to-right and right-to-left manners under the guidance of previously generated forward and backward outputs. We apply our bidirectional attention network to replace the self-attention network in its decoder part, and illustrate the overall architecture in Figure 2. Next, we will present those two bidirectional attention models and integrate them into the decoder of Transformer.

Bidirectional Scaled Dot-Product Attention

Figure 3 (left) shows our particular attention. The input consists of queries ($[\vec{Q}; \overleftarrow{Q}]$), keys ($[\vec{K}; \overleftarrow{K}]$) and values ($[\vec{V}; \overleftarrow{V}]$) which are all concatenated by forward (L2R) states and backward (R2L) states. The new forward states \vec{H}_j and backward states \overleftarrow{H}_j can be obtained by bidirectional dot-product scaled attention. For new forward states \vec{H}_j , it can be calculated as:

$$\vec{H}_j^f = \text{ATT}(\vec{Q}_j, \vec{K}_{\leq j}, \vec{V}_{\leq j}) = \text{softmax}\left(\frac{\vec{Q}_j \vec{K}_{\leq j}^T}{\sqrt{d_k}}\right) \vec{V}_{\leq j} \quad (4)$$

$$\overleftarrow{H}_j^b = \text{ATT}(\overleftarrow{Q}_j, \overleftarrow{K}_{\leq j}, \overleftarrow{V}_{\leq j}) = \text{softmax}\left(\frac{\overleftarrow{Q}_j \overleftarrow{K}_{\leq j}^T}{\sqrt{d_k}}\right) \overleftarrow{V}_{\leq j}$$

where \vec{H}_j^f is obtained by conventional scaled dot-product attention as introduced in Equation 2, and \overleftarrow{H}_j^b contains the attentional future information from R2L decoding. Then we

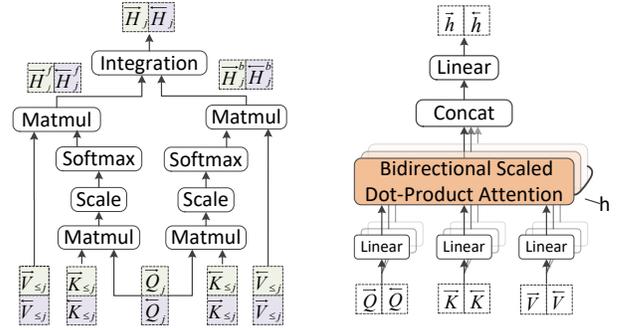


Figure 3: (left) Bidirectional Scaled Dot-Product Attention operates on forward (L2R) and backward (R2L) queries Q , keys K , values V . (right) Bidirectional Multi-Head Intra-Attention consists of several attention layers in parallel.

use a linear interpolation method to integrate the forward information \vec{H}_j^f and backward information \overleftarrow{H}_j^b :

$$\vec{H}_j = \text{Integration}(\vec{H}_j^f, \overleftarrow{H}_j^b) = \vec{H}_j^f + \lambda * \overleftarrow{H}_j^b \quad (5)$$

where λ is a hyper-parameter decided by the performance on development set.

For R2L decoding, similar to the calculation of forward hidden states \vec{H}_j , the backward hidden states \overleftarrow{H}_j can be computed as follows.

$$\overleftarrow{H}_j^f = \text{ATT}(\overleftarrow{Q}_j, \overleftarrow{K}_{\leq j}, \overleftarrow{V}_{\leq j})$$

$$\overleftarrow{H}_j^b = \text{ATT}(\overleftarrow{Q}_j, \vec{K}_{\leq j}, \vec{V}_{\leq j}) \quad (6)$$

$$\overleftarrow{H}_j = \text{Integration}(\overleftarrow{H}_j^f, \overleftarrow{H}_j^b)$$

where $\text{Integration}(\cdot)$ is the same as introduced in Equation 5. We refer to the whole procedure formulated in Equation 4-6 as BSDPA(\cdot).

$$[\vec{H}_j; \overleftarrow{H}_j] = \text{BSDPA}([\vec{Q}_j; \overleftarrow{Q}_j], [\vec{K}_{\leq j}; \overleftarrow{K}_{\leq j}], [\vec{V}_{\leq j}; \overleftarrow{V}_{\leq j}]) \quad (7)$$

It is worth noting that \vec{H}_j and \overleftarrow{H}_j can improve each other and be calculated in parallel.

Bidirectional Multi-Head Intra-Attention

Different from the mask multi-head attention (Equation 3), we can obtain the new forward and backward hidden states simultaneously, as shown in Figure 3 (right), where i -th attention head with j -th target token can be computed using BSDPA(\cdot):

$$\text{head}_{i,j} = [\vec{h}_{i,j}; \overleftarrow{h}_{i,j}] = \text{BSDPA}([\vec{Q}_j; \overleftarrow{Q}_j]W_i^Q, [\vec{K}_{\leq j}; \overleftarrow{K}_{\leq j}]W_i^K, [\vec{V}_{\leq j}; \overleftarrow{V}_{\leq j}]W_i^V) \quad (8)$$

where W_i^Q, W_i^K and W_i^V are parameter matrices, which are the same as standard multi-head attention introduced in Equation 3. By contrast, bidirectional multi-head inter-attention is composed of two standard multi-head attention models, which do not interact with each other.

Integrating Bidirectional Attention into Decoder

We use our bidirectional attention network to replace the multi-head attention in the decoder part, as demonstrated in Figure 2. For each layer in bidirectional decoder, the first sub-layer is the **bidirectional multi-head intra-attention**

(BiAtt^{Intra}) network¹ which is capable of combining history and future information:

$$[\vec{s}_d^l; \overleftarrow{s}_d^l] = \text{BiAtt}^{\text{Intra}}([\vec{s}_d^{l-1}; \overleftarrow{s}_d^{l-1}], [\vec{s}^{l-1}; \overleftarrow{s}^{l-1}], [\vec{s}^{l-1}; \overleftarrow{s}^{l-1}]) \quad (9)$$

where s^l denotes l -layer hidden states or embedding vectors when $l=0$, and subscript d denotes the decoder-informed intra-attention representation.

The second sub-layer is the **bidirectional multi-head inter-attention** (BiAtt^{Inter}) which integrates the representation of the corresponding source sentence by performing left-to-right and right-to-left decoding attention respectively, as shown in Figure 2.

$$[\vec{s}_e^l; \overleftarrow{s}_e^l] = \text{BiAtt}^{\text{Inter}}([\vec{s}_d^l; \overleftarrow{s}_d^l], [h^N; h^N], [h^N; h^N]) \quad (10)$$

where e denotes the encoder-informed inter-attention representation, and h^N is the source hidden state of top layer.

The third sub-layer is a position-wise fully connected feed-forward neural network: $[\vec{s}_j^l; \overleftarrow{s}_j^l] = \text{FFN}([\vec{s}_e^l; \overleftarrow{s}_e^l])$.

Finally, we employ a linear transformation and softmax activation to compute the probability of the j -th tokens based on N -layer $s_j^N = [\vec{s}_j^N; \overleftarrow{s}_j^N]$, namely the final hidden states of forward and backward decoding.

$$\begin{aligned} p(\vec{y}_j | \vec{y}_{<j}, \overleftarrow{y}_{<j}, x, \theta) &= \text{softmax}(\vec{s}_j^N W) \\ p(\overleftarrow{y}_j | \overleftarrow{y}_{<j}, \vec{y}_{<j}, x, \theta) &= \text{softmax}(\overleftarrow{s}_j^N W) \end{aligned} \quad (11)$$

where W denotes the weight matrix and θ is the shared parameters for L2R and R2L decoding.

3.3 Training and Inference

Training. Given a parallel sentence pair (x, y) , we design a smart strategy to enable synchronous bidirectional generation within a decoder. We first divide the output sentence (y) into two halves and reverse the second half. Second, we separately add the special labels ($\langle l2r \rangle$ and $\langle r2l \rangle$) at the beginning of each half sentence (\vec{y} and \overleftarrow{y}) to guide generating tokens from left to right or right to left. Finally, we propose a smoothing model to better connect both directional generational results. As shown in Figure 4, if the output length is odd, we add the additional tag ($\langle null \rangle$) before $\langle eos \rangle$ in forward or backward sentence randomly. In other words, our model is capable of generating a null word when necessary. Following previous work [Gu *et al.*, 2017; Wang *et al.*, 2018], we also use knowledge distillation techniques [Kim and Rush, 2016] to train our model. 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 likeli-

¹Note that we follow Vaswani *et al.* [Vaswani *et al.*, 2017] to use residual connection and layer normalization in each decoder sub-layer, which are omitted in the presentation for simplicity.

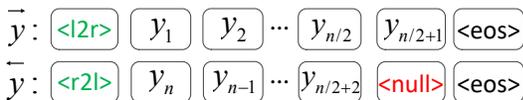


Figure 4: The smoothing model introduced to connect L2R and R2L results smoothly. When the output sentence has odd tokens, we randomly insert $\langle null \rangle$ which means null word and can be removed in postprocessing.

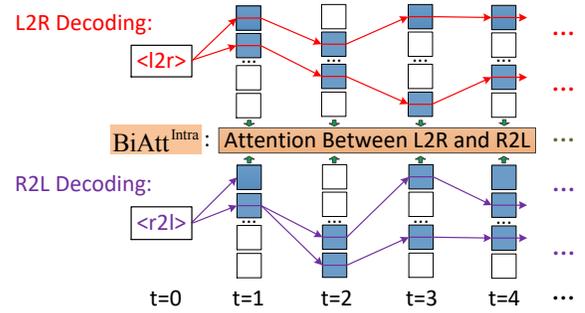


Figure 5: The bidirectional beam search process of our proposed model, which produces tokens from left-to-right and right-to-left simultaneously, under the guidance of two special labels ($\langle l2r \rangle$ and $\langle r2l \rangle$). By using bidirectional attention model, left-to-right and right-to-left decoding can help and interact with each other.

hood of the training data:

$$\begin{aligned} \mathbb{L}(\theta) &= \frac{1}{Z} \sum_{z=1}^Z \sum_{j=1}^{n/2} \{ \log p(\vec{y}_j^{(z)} | \vec{y}_{<j}^{(z)}, \overleftarrow{y}_{<j}^{(z)}, x^{(z)}, \theta) \\ &\quad + \log p(\overleftarrow{y}_j^{(z)} | \overleftarrow{y}_{<j}^{(z)}, \vec{y}_{<j}^{(z)}, x^{(z)}, \theta) \} \end{aligned} \quad (12)$$

Inference. Once the proposed model is trained, we employ a simple bidirectional beam search algorithm to predict the output sequence. As illustrated in Figure 5, with two special start tokens which are optimized during the training process, we let half of the beam to keep decoding from left to right, and allow the other half beam to decode from right to left. The blue blocks denote the ongoing expansion of the hypothesis and decoding terminates when the end-of-sentence flag is predicted. More importantly, by using the bidirectional multi-head intra-attention, the two decoding manners can help and interact with each other in one beam-search process. Alternatively, we can also use greedy search to our model.

4 Application to Neural Machine Translation

We use BLEU [Papineni *et al.*, 2002] to evaluate the proposed model on translation tasks.

4.1 Setup

We verify our model on three translation datasets of different sizes: WMT14 English-German² (En \Rightarrow De), NIST Chinese-English³ (Ch \Rightarrow En), WMT16 English-Romanian⁴ (En \Rightarrow Ro), whose training sets consist of 4.5M, 2.0M, 0.6M sentence pairs, respectively. We tokenize the corpora using a script from Moses [Koehn *et al.*, 2007] and segment each word into subword units using BPE [Sennrich *et al.*, 2016b]. We use 37K and 40K shared BPE tokens for En \Rightarrow De and En \Rightarrow Ro respectively. For En \Rightarrow De, we use newstest2013 as the validation set and newstest2014 as the test set. For Ch \Rightarrow En, we utilize BPE to encode Chinese and English respectively, and limit the source and target vocabularies to the most frequent 30K tokens. We use NIST 2006 as the validation set, NIST

²<http://www.statmt.org/wmt14/translation-task.html>

³The corpora include LDC2000T50, LDC2002T01, LDC2002E18, LDC2003E07, LDC2003E14, LDC2003T17 and LDC2004T07.

⁴<http://www.statmt.org/wmt16/translation-task.html>

System	Architecture	English-German		Chinese-English		English-Romanian	
		Quality	Speed	Quality	Speed	Quality	Speed
Existing NMT systems							
[Gu <i>et al.</i> , 2017]	NAT	17.35	N/A	-	-	26.22	15.6×
	NAT (s=100)	19.17	N/A	-	-	29.79	2.36×
[Lee <i>et al.</i> , 2018]	D-NAT	12.65	11.71×	-	-	24.45	16.03×
	D-NAT (adaptive)	18.91	1.98×	-	-	29.66	5.23×
[Kaiser <i>et al.</i> , 2018]	LT	19.80	3.89×	-	-	-	-
	LT (s=100)	22.50	N/A	-	-	-	-
[Wang <i>et al.</i> , 2018] (beam search)	SAT (K=2)	26.90	1.51×	39.57	1.69×	-	-
	SAT (K=6)	24.83	2.98×	35.32	3.18×	-	-
[Wang <i>et al.</i> , 2018] (greedy search)	SAT (K=2)	26.09	1.70×	38.37	1.71×	-	-
	SAT (K=6)	23.93	4.57×	33.75	4.70×	-	-
Our NMT systems							
This work (beam search)	Transformer	27.06	1.00×	46.56	1.00×	32.28	1.00×
	Transformer (R2L)	26.71	1.02×	44.63	0.94×	32.29	0.98×
	Our Model	27.45	1.38×	47.82	1.41×	33.02	1.43×
This work (greedy search)	Transformer	26.23	1.00×	44.63	1.00×	31.71	1.00×
	Transformer (R2L)	25.38	0.97×	43.68	0.98×	31.19	1.04×
	Our Model	27.22	1.61×	47.50	1.51×	32.82	1.46×

Table 1: Translation quality (BLEU) and speed on official test sets. Translation speed is measured on the amount of translated sentences in one second. For comparison, we also list results reported by Gu *et al.* [2017]; Lee *et al.* [2018]; Kaiser *et al.* [2018]; Wang *et al.* [2018]. Note that we and SAT use different size corpus and different preprocessing methods for Chinese-English translation. Although the non-autoregressive or semi-autoregressive NMT models have greater potential in speedup decoding than ours, the major drawback is translation quality degradation. By making full use of the history information and future information, our SBSG model can get a significant BLEU improvement ($p < 0.01$) than autoregressive, semi-autoregressive and non-autoregressive models.

2003-2005 as our test sets. For En \Rightarrow Ro, we use newsdev-2016 and newstest-2016 as development and test sets.

We implement the proposed model based on the tensor2tensor⁵ toolkit. 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 $\epsilon_{ls}=0.1$. We use three GPUs to train En \Rightarrow De and one GPU for the other two language pairs. For evaluation, we use beam search with a beam size of $k=4$ and length penalty $\alpha=0.6$. Besides, we use 6 encoder and decoder layers, 512 hidden size, 8 attention-heads, 2048 feed-forward inner-layer dimensions.

4.2 Results and Analysis

Parameters. NAT [Gu *et al.*, 2017] adopts encoder-decoder architecture with additional fertility predictor model. D-NAT [Lee *et al.*, 2018] has two decoders and needs more parameters than conventional Transformer. Our bidirectional NMT model uses one single encoder-decoder model, which can predict the target tokens in left-to-right and right-to-left manners simultaneously. Hence, our SBSG model does not increase any parameters except for a hyper-parameter λ compared to the standard Transformer.

Inference Speed. As shown in Table 1, the proposed SBSG model is capable of decoding approximately 1.4× faster than autoregressive Transformer with beam search in three translation tasks. Besides, our model obtains 1.61× (En \Rightarrow De), 1.51× (Ch \Rightarrow En), and 1.46× (En \Rightarrow Ro) speedup

than Transformer in greedy search. As a compromise solution between autoregressive and non-autoregressive models, the speed of our model is relatively slower than NAT, D-NAT, and LT [Kaiser *et al.*, 2018]. Besides, our proposed model is capable of obtaining comparable translation speed compared to SAT [Wang *et al.*, 2018] with K=2.

Translation Quality. Table 1 shows translation performance of En \Rightarrow De, Ch \Rightarrow En, and En \Rightarrow Ro translation tasks. The proposed model behaves better than NAT, D-NAT, LT in all test datasets. In particular, our model with beam search significantly outperforms NAT, D-NAT, and LT by 8.28, 8.54 and 4.95 BLEU points in large-scale English-German translation, respectively. Although the SAT has a faster decoding speed than the SBSG model when K becomes bigger, it suffers from the translation quality degradation relative to the autoregressive NMT. Compared to autoregressive Transformer, our proposed model with beam search is able to behave better in terms of both decoding speed and translation quality. Furthermore, our model with greedy search does not only outperform autoregressive Transformer by 0.99, 2.87 and 1.11 BLEU points of translation quality in En \Rightarrow De, Ch \Rightarrow En and En \Rightarrow Ro respectively, but also significantly speedups the decoding of conventional Transformer.

Length Analysis. We follow Bahdanau *et al.* [2015] to group sentences of similar lengths together, and compute a BLEU score and the averaged length of translations per group. Figure 6 shows that the performance of Transformer and Transformer (R2L) drops rapidly when the length of the input sentence increases. Our SBSG model alleviates this problem by generating a sequence from both sides to the middle, which in general encourages the model to produce more

⁵<https://github.com/tensorflow/tensor2tensor>

Source	徽标由乘风破浪西行的帆船、翻滚的浅蓝色弧形水纹浪花和文字”郑和下西洋 600 周年 1405 - 2005”组成。
Reference	the logo was made up of a westbound sailboat braving the wind and waves , churning arc - shaped spindrifts in light blue color , and words that say ” zheng he ’s 600th anniversary to the west , 1405 - 2005 . ”.
L2R	the logo is composed of sailboats that break the wind and break the waves , rolling light blue water wave flowers .
R2L	1405-2005 of zheng he , the 600th anniversary of the west .
SBSG	the logo is composed of sailing boat by wind and waves , rolling light blue surpo- shaped flowers , and words ” zheng to the west , 1405 - 2005 ” .

Table 2: A Chinese-English translation example of baselines and our proposed model. Our model can alleviate the unbalanced output problems [Liu *et al.*, 2016b] by generating a sentence from both sides to the middle.

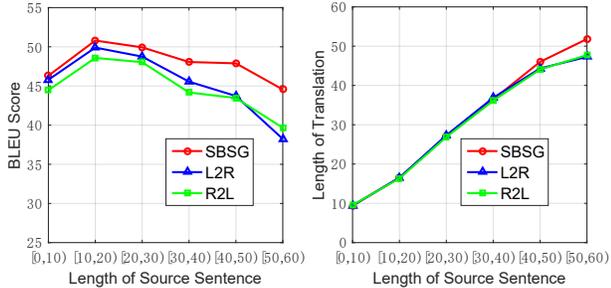


Figure 6: Length Analysis - Performance of the generated translations with respect to the lengths of the source sentences. The proposed SBSG model can alleviate under-translation by producing longer translation on long sentences.

accurate and long sentences.

Case Study. In Table 2, we present a translation example from NIST Chinese-English. Liu *et al.* [2016b] show that L2R model produces the translation with good prefix and sometimes omits the second half sentence, but L2R model usually generates the generation with better suffixes. Our results confirm these findings. The proposed SBSG model can alleviate the errors by generating sequences from both sides to the middle.

5 Application to Text Summarization

We further verify the effectiveness of our proposed SBSG model on text summarization, which is another real-world application that encoder-decoder framework succeeds [Rush *et al.*, 2015].

5.1 Setup

Abstractive sentence summarization aims to provide a title-like summary for a long sentence. We conduct text summarization experiments on English Gigaword dataset⁶. The parallel corpus is produced by pairing the first sentence and the headline in the news article with some heuristic rules. The extracted corpus contains about 3.8M sentence-summary pairs for the training set and 189K examples for the development set. We employ a shared vocabulary of about 90K word types and use DUC 2004 used by Rush *et al.* [2015] as our test set. The model structure is the same as that used in neural machine translation. We employ ROUGE as our evaluation metric, which is all widely adopted evaluation metric for text summarization.

⁶<https://github.com/harvardnlp/sent-summary>

DUC2004	RG-1	RG-2	RG-L	Speed
ABS [‡]	26.55	7.06	22.05	-
Feats2s [‡]	28.35	9.46	24.59	-
Selective-Enc [‡]	29.21	9.56	25.51	-
Transformer	28.09	9.52	24.91	1.00×
SBSG (beam)	28.77	10.11	26.11	1.48×
SBSG (greedy)	28.70	9.88	25.93	2.09×

Table 3: ROUGE recall evaluation results on DUC 2004 test set. For comparison, we also list results reported by Rush *et al.* [2015]; Nalapat *et al.* [2016]; Zhou *et al.* [2017]. Results with [‡] mark are taken from the corresponding papers. Our proposed SBSG model significantly outperforms the conventional Transformer model in terms of both decoding speed and generation quality.

5.2 Results and Analysis

In Table 3, we report the ROUGE score and speed for DUC 2004 test set. Experiments show that the generation quality of our proposed model is on par with the state-of-the-art text summarization models. We observe approximately 1.5× faster decoding than the autoregressive Transformer while achieving better generation quality. Specially, our model with beam search (greedy search) is capable of decoding 1.64× (2.26×) faster than conventional Transformer on English Gigaword test set.

6 Conclusions

In this work, we propose a novel SBSG model that performs bidirectional decoding simultaneously and interactively. Instead of producing output sentence token by token, the proposed model makes decoding much more parallelizable and generates two tokens at each time step. We extensively evaluate the proposed SBSG model on neural machine translation (En⇒De, Ch⇒En, and En⇒Ro) and text summarization (English Gigaword) tasks. Different from previous non-autoregressive models [Gu *et al.*, 2017; Lee *et al.*, 2018; Kaiser *et al.*, 2018] which suffer from serious quality degradation, our SBSG model achieves a significant improvement in both generation quality and decoding speed compared to the state-of-the-art autoregressive Transformer.

Acknowledgments

The research work described in this paper has been funded by the National Key Research and Development Program of China under Grant No. 2016QY02D0303 and the Natural Science Foundation of China under Grant No. U1836221 and 61673380. This work is also supported by grants from NVIDIA NVAIL program.

References

- [Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.
- [Devlin, 2017] Jacob Devlin. Sharp models on dull hardware: Fast and accurate neural machine translation decoding on the cpu. In *EMNLP*, pages 2820–2825, 2017.
- [Finch and Sumita, 2009] Andrew Finch and Eiichiro Sumita. Bidirectional phrase-based statistical machine translation. In *EMNLP*, pages 1124–1132, 2009.
- [Gehring *et al.*, 2017] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. Convolutional sequence to sequence learning. In *ICML*, 2017.
- [Gu *et al.*, 2017] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.
- [Kaiser *et al.*, 2018] Łukasz Kaiser, Aurko Roy, Ashish Vaswani, Niki Pamar, Samy Bengio, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. *arXiv preprint arXiv:1803.03382*, 2018.
- [Kim and Rush, 2016] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *EMNLP*, 2016.
- [Koehn *et al.*, 2007] 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. Moses: Open source toolkit for statistical machine translation. In *ACL*, 2007.
- [Lee *et al.*, 2018] Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *EMNLP*, pages 1173–1182, 2018.
- [Li *et al.*, 2018] Haoran Li, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization. In *COLING*, 2018.
- [Liu *et al.*, 2016a] Lemao Liu, Andrew M Finch, Masao Utiyama, and Eiichiro Sumita. Agreement on target-bidirectional lstms for sequence-to-sequence learning. In *AAAI*, 2016.
- [Liu *et al.*, 2016b] Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. Agreement on target-bidirectional neural machine translation. In *NAACL*, 2016.
- [Liu *et al.*, 2018] Yuchen Liu, Long Zhou, Yining Wang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. A comparable study on model averaging, ensembling and reranking in nmt. In *NLPCC*, pages 299–308, 2018.
- [Mi *et al.*, 2016] Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. Vocabulary manipulation for neural machine translation. In *ACL*, pages 124–129, 2016.
- [Oord *et al.*, 2017] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv preprint arXiv:1711.10433*, 2017.
- [Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002.
- [Rush *et al.*, 2015] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *EMNLP*, 2015.
- [Sennrich *et al.*, 2016a] Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for wmt 16. In *WMT*, 2016.
- [Sennrich *et al.*, 2016b] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, pages 1715–1725, 2016.
- [Serdyuk *et al.*, 2018] Dmitriy Serdyuk, Nan Rosemary Ke, Alessandro Sordani, Adam Trischler, Chris Pal, and Yoshua Bengio. Twin networks: Matching the future for sequence generation. In *ICLR*, 2018.
- [Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014.
- [Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.
- [Vinyals *et al.*, 2015] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.
- [Wang *et al.*, 2018] Chunqi Wang, Ji Zhang, and Haiqing Chen. Semi-autoregressive neural machine translation. *arXiv preprint arXiv:1808.08583*, 2018.
- [Watanabe and Sumita, 2002] Taro Watanabe and Eiichiro Sumita. Bidirectional decoding for statistical machine translation. In *COLING*, 2002.
- [Xu *et al.*, 2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. *Computer Science*, pages 2048–2057, 2015.
- [Zhang *et al.*, 2018a] Biao Zhang, Deyi Xiong, and jinsong Su. Accelerating neural transformer via an average attention network. In *ACL*, pages 1789–1798, 2018.
- [Zhang *et al.*, 2018b] Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. Asynchronous bidirectional decoding for neural machine translation. In *AAAI*, 2018.
- [Zhou *et al.*, 2017] Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. Selective encoding for abstractive sentence summarization. In *ACL*, pages 1095–1104, 2017.
- [Zhou *et al.*, 2019] Long Zhou, Jiajun Zhang, and Chengqing Zong. Synchronous bidirectional neural machine translation. In *TACL*, pages 91–105, 2019.