Abstract

Simultaneous Neural Machine Translation (SimulNMT) generates the output before the entire input sentence is available and only uses the unidirectional attention from left-to-right so that its decoding highly relies on future forecast according to word ordering rules. However, it is utopian that the word order strictly obeys the grammar rules in a language, especially in oral. To address the mismatch between SimulNMT expecting strict word order and free word order in real scenario, we propose a bidirectional modeling. In detail, we train another backward model where the input sentence is from right-to-left and keep the target sentence from left-to-right. Then we join this backward model into the standard forward SimulNMT model during decoding. This strategy enhances the robustness of SimulNMT and empowers the model to be more adaptable for the inconstant word ordering phenomenon. Experiments show that our method brings improvement over the strong baselines.

1 Introduction

Neural Machine Translation (NMT), built on the encoder-decoder framework has achieved advanced translation performance in recent years other than the traditional statistical machine translation (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). Inside an NMT model, the encoder encodes a source input sentence, and the decoder generates the target language sentence by iteratively predicting the output token according to the entire input with the partially decoded output so far. However, these offline models mentioned above are not well adaptable for real-time speech-to-speech interpretation, such as international conferences, symposiums, and business. Thus, online (or simultaneous) NMT is quite desirable for such scenarios, which starts the decoding process right after reading the first few words of the source sentence instead of waiting for the end.

SimulNMT has caused widespread concern in the NMT community recently (Cho and Esipova, 2016; Jaitly et al., 2016; Dalvi et al., 2018; Ma et al., 2019; Zheng et al., 2019a; Zhang et al., 2019; Arivazhagan et al., 2019; Ma et al., 2020; Ren et al., 2020; Elbayad et al., 2020). Ma et al. (2019) propose a popular wait-k decoding algorithm where the decoding process is always k words after the source input instead of single read-writes. This simple approach guarantees the translation quality and controls the translation delay at the same time. For dynamic online decoding, reinforcement learning (RL) and imitation learning (IL) are also used to optimize the read/write policy (Grissom II et al., 2014; Luo et al., 2017; Gu et al., 2017; Press and Smith, 2018; Zheng et al., 2019b).

However, all of the methods are decoded from left-to-right without the future information, ignoring that word order is flexible so that the resulting translations cannot always obey the grammar rules in practical use, especially in oral. The SimulNMT performance may be greatly hindered by the mismatch about the word order forecast between the requirement of SimulNMT and the actual scenario in linguistics..

To address such a mismatch issue in the current SimulNMT, we propose a bidirectional modeling strategy in this work. In detail, we train another backward model, in contrast to the forward model shown in the Figure 1, which inputs the source sentence from right-to-left and keeps the target sentence left-to-right order. Then we joint this backward model into the forward model during decoding. This decoding policy enhances the robustness of SimulNMT and allows the model to be more adaptable for the inconstant word order phenomenon. Experiments show that our method significantly improves the translation performance
over the strong baselines.

2 Backward Modeling as Future Forecasting

2.1 Problem Formalization

Given a source sentence \( x = \{x_1, \ldots, x_t, \ldots, x_{L_x}\} \) in the document to be translated and a target sentence \( y = \{y_1, \ldots, y_t, \ldots, y_{L_y}\} \), we denote \( x_{\leq t} \) as the a substring of \( x \) containing words \( \{x_1, \ldots, x_t\} \), and similarly for \( y_{\leq t} \) and \( y_{< t} \). The NMT model computes the probability of translation from the source sentence to the target sentence word by word:

\[
P(y|x) = \prod_{t=1}^{L_y} P(y_t|y_{< t}, x), \tag{1}
\]

In this paper, we focus on the SimulNMT model based on the Transformer (Vaswani et al., 2017) which contains an encoder and a decoder and respectively processes the source and target sentences. Both are composed of a stack of \( N \) (usually equal to 6) identical layers. The critical component is multi-headed attention, which concatenates the outputs from multiple attention heads.

2.2 Simultaneous Neural Machine Translation (SimulNMT)

SimulNMT starts decoding the translation before the entire input sentence is available. Formally, we use \( z_t \) to represent the number of source tokens read when decoding \( y_t \). In the SimulNMT model, the decoder predicts \( y_t \) by considering the first \( z_t \) source states, and each source state only encodes the information from the \( z_{t-1} \) source tokens read so far.

Unidirectional Transformer Encoder In the most encoder-decoder model, encoding the source tokens at a given position includes information from the past and future time-steps. However, the encoder has to be recomputed when the new source token is inputted. To reduce the cost of re-encoding the source sequence, Elbayad et al. (2020) propose unidirectional encoders for SimulNMT by masking the self-attention and only consider the previous time-steps. In this way, source sentences are encoded once without updating the encoder states at each time step.

**Wait-\( k \) Strategy** Human simultaneous interpretation usually starts translating a few seconds after the speakers’ speech and finishes a few seconds accordingly after the speaker finishes. Inspired by this, Ma et al. (2019) present a wait-\( k \) policy, which first waits for the \( k \) source tokens and then translates simultaneously with the rest of the source sentence. When \( k = \infty \), the full source sentence is read before decoding. For a wait-\( k \) decoding path, \( z_t = \min\{k + t - 1, L_x\} \). The SimulNMT model computes the probability with regard to the single wait-\( k \) decoding path \( z^k \):

\[
P(y|x, z^k) = \prod_{t=1}^{L_y} P(y_t|y_{< t}, x_{\leq z_t^k}, z_{< t}^k) \tag{2}
\]

The wait-\( k \) strategy is most effective when trained for the specific \( k \) (Zheng et al., 2019b). However, it requires training models individually for each potential value of \( k \) for translation.

2.3 Backward Modeling

Current SimulNMT methods translate the output tokens word by word from left to right, which is denoted as *forward* model. Considering the flexible word order phenomenon, we train another *backward* model by the contrast, which takes the source sentence as input from right to left, and keeps the target sentence in standard left-to-right order illustrated in Figure 2. It is worth noting that we adopt unidirectional self-attention in the *forward* modeling as in the standard SimulNMT, but a bidirectional self-attention in the *backward* modeling.
We briefly denote English, German, Vietnamese as En, De, and Vi respectively and conduct our simul-

<table>
<thead>
<tr>
<th>Task</th>
<th>wait-k</th>
<th>Zheng et al.</th>
<th>Elbayad et al.</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWSLT14 En→ De</td>
<td>26.74</td>
<td>-</td>
<td>26.40</td>
<td>27.39 (↑0.65)</td>
</tr>
<tr>
<td>IWSLT14 De→ En</td>
<td>30.15</td>
<td>30.17</td>
<td>30.48</td>
<td>32.13 (↑1.65)</td>
</tr>
<tr>
<td>IWSLT15 En→ Vi</td>
<td>28.31</td>
<td>-</td>
<td>29.19</td>
<td>29.98 (↑0.79)</td>
</tr>
<tr>
<td>IWSLT15 Vi→ En</td>
<td>21.89</td>
<td>-</td>
<td>22.32</td>
<td>23.20 (↑0.88)</td>
</tr>
</tbody>
</table>

Table 1: The results of our proposed models for $k_{train} = k_{eval} = 7$.

Figure 2: The framework of our proposed model using wait-$k$ strategy and $k = 3$

On the one hand, backward modeling allows the whole to be exposed to a new order, and on the other hand, bidirectional modeling can bring more comprehensive features as an aid to normal forward SimulNMT. The probability of the translation in the backward model is calculated as follows:

$$P(y|x, z^k) = \prod_{t=1}^{L_y} P(y_t|y_{<t}, x_{(t-L_y+1)^t}, z^k_{<t})$$

(3)

Then we train an ensemble model to joint this backward model output into the forward model during decoding.

$$P_{ensemble} = w_1 \otimes P_{backward} + w_2 \otimes P_{forward}$$

(4)

where $w_1$ and $w_2$ respectively mean the weights of the backward and the forward models. In our work, we set $w_1 = 0.1$ and $w_2 = 1.0$ based on the preliminary experiments.

3 Experiments

We first evaluate models trained with different wait-$k$ decoding paths on the IWSLT14 En→De datasets. Figure 3 presents the performance of the models trained with $k_{train} \in \{1, 3, 5, 7, 9\}$ on these two datasets. Each curve with specified color represents each trained model, which is evaluated across different wait-$k$ decoding paths with
we perform better by a great margin. The evaluation results on multiple benchmarks show that our proposed method achieves well on other evaluation paths. Like most simul- NMT models using wait-\(k\) policy, the performance drops when far from the training path, for example, when \(k_{\text{train}} = 1\) and \(k_{\text{eval}} = 9\).

We also compare our method with the related work on IWSLT datasets when \(k_{\text{train}} = k_{\text{eval}} = 7\). As shown in Table 1, our proposed method achieves the highest BLEU scores than the baselines and related works. Especially on IWSLT14 En→De, we perform better by a great margin. The evaluation results on multiple benchmarks show that our approach can obtain better scores than the baseline, and the latency is not affected much. This shows that the performance of SimulNMT models can be effectively improved through additional model design. We empirically verified that using our proposed bidirectional modeling is simple and effective.

4 Ablation Study

To investigate the importance of the forward model and backward model, we provide three groups of ablation study: (1) only with the forward model, (2) only with the backward model, and (3) the ensemble model with two forward models with a different seed. We work on the IWSLT14 En→De task and study the effect to wait-\{-3, 5, 7\}. The results are shown in Table 2, and the latency metrics (AP and AL) are not significantly influenced. The BLEU score drops when removing any feature, which indicates that they all benefit the model. Specifically, the forward model plays the most critical role in our model. This reveals that due to the use of uni-directional attention in the forward modeling of SimulNMT, although real-time efficiency is satisfied, translation quality suffers from a negative impact. And with an additional feature from backward modeling added for enhancing the forward modeling, it indeed enhances the model’s ability to adapt to the flexible order without affecting the latency.

5 Conclusion

In this work, we proposed a bidirectional modeling strategy for simultaneous neural machine translation. Motivated by the observation that the word order is free in practical use, while SimulNMT expects strict word order, we train a backward decoding model to let wait-\(k\) forecast the future information. Then we fuse the backward model into the forward model for ensemble decoding. Experiments on four translation tasks indicate the effectiveness of our model design. Experimental results demonstrate that this method is simple and effective. For future work, we will enhance this model performance by jointing more considerable auxiliary models besides the backward model.

<table>
<thead>
<tr>
<th>(k = 3)</th>
<th>(k = 5)</th>
<th>(k = 7)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Our model</strong></td>
<td><strong>Our model</strong></td>
<td><strong>Our model</strong></td>
</tr>
<tr>
<td>BLEU</td>
<td>AP</td>
<td>AL</td>
</tr>
<tr>
<td>- only with <strong>forward</strong> model</td>
<td>23.59</td>
<td>0.66</td>
</tr>
<tr>
<td>- only with <strong>backward</strong> model</td>
<td>23.45</td>
<td>0.65</td>
</tr>
<tr>
<td>- with two different <strong>forward</strong> model</td>
<td>8.54</td>
<td>0.66</td>
</tr>
<tr>
<td>- only with <strong>backward model</strong></td>
<td>22.68</td>
<td>0.65</td>
</tr>
</tbody>
</table>

Table 2: Ablation Study in IWSLT14 En→De
References


3787–3796, Online. Association for Computational Linguistics.


