Toward Neural Phrase-based Machine Translation

Po-Sen Huang [*1] **Chong Wang** [*1] **Dengyong Zhou** [1] **Li Deng**[†2]

Abstract

In this paper, we propose Neural Phrase-based Machine Translation (NPMT). Our method explicitly models the phrase structures in output sequences through Sleep-WAke Networks (SWAN), a recently proposed segmentation-based sequence modeling method. To alleviate the monotonic alignment requirement of SWAN, we introduce a new layer to perform (soft) local reordering of input sequences. Our experiments show that NPMT achieves state-of-the-art results on IWSLT 2014 German-English translation task without using any attention mechanisms. We also observe that our method produces meaningful phrases in the output language.

1. Introduction

Human languages often exhibit strong compositional patterns. For example, consider understanding the following sentence, “machine learning is part of artificial intelligence.” It may become easier to comprehend if we segment it as “[machine learning] [is] [part of] [artificial intelligence]”, where the words in the bracket ‘[]’ are often regarded as “phrases”. These phrases have their own meanings, and can be reused in other contexts.

In this paper, we develop a neural machine translation method that explicitly models phrases on the output language. Traditional statistical phrase-based machine translation approaches have been shown to consistently outperform word-based ones (Koehn et al., 2003; Koehn, 2009; Lopez, 2008). On the other hand, modern neural machine translation (NMT) methods (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015) do not have an explicit treatment on phrases, but they still work surprisingly well. Our Neural Phrase-based Machine Translation (NPMT) method tries to explore advantages from both kingdoms. It is built upon Sleep-WAke Networks (SWAN), a segmentation-based sequence modeling technique (Wang et al., 2017). Here, segments are regarded as phrases in target sequences. However, SWAN requires monotonic alignments between inputs and outputs. That is often not the case in machine translation. To fix this issue, we introduce a new layer below SWAN to perform (soft) local reordering on input sequences. Preliminary experiments show that NPMT outperforms attention-based NMT baselines in terms of the BLEU score (Papineni et al., 2002).

This paper is organized as follows. Section 2 presents the neural phrase-based machine translation model. Section 3 demonstrates the usefulness of our approach on IWSLT 2014 German-to-English translation task. We conclude our work with some discussions in Section 4.

2. Neural phrase-based machine translation

We first review SWAN, and then show a reordering model to alleviate its monotonic alignments requirement. NPMT is built upon SWAN and the reordering module.[1]

2.1. Modeling phrases with SWAN

SWAN models all valid output segmentations as well as the monotonic alignments between the output segments and the input sequence. Empty segments are allowed in the output segmentations. SWAN does not make any assumption on the lengths of input or output sequence.

Assume input sequence is $x_{1:T'}$ and output sequence is $y_{1:T}$. $\mathcal{S}_y$ denotes the set containing all valid segmentations of $y_{1:T}$, where the number of segments in any segmentation is always T', the input sequence length. Empty segments are allowed to ensure that we can correctly align segment a_t to input element x_t. Otherwise, we might not always have a valid alignment for the input and output pair. See Figure 1 for an example of the emitted segmentation of $y_{1:T}$. The probability of the sequence $y_{1:T}$ is defined as the sum

[*] Equal contribution [1] Microsoft Research [2] Citadel Securities LLC. [†] Work performed when the author was with Microsoft. Correspondence to: Po-Sen Huang, Chong Wang, Dengyong Zhou <{pshuang,chowang,denzho}@microsoft.com>.

Proceedings of the ICML 17 Workshop on Deep Structured Prediction, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

[1] In NPMT, we explicitly consider output segmentations using SWAN while input segmentations are implicitly modeled via empty segments in SWAN. We leave the option of exploring explicit input segmentations as future work.

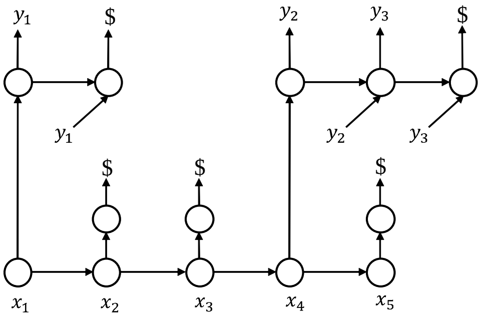

Figure 1. Courtesy to Wang et al. (2017). Symbol \$ indicates the end of a segment. SWAN emits one particular segmentation of $y_{1:3} = \pi(a_{1:5})$ with x_1 waking (emitting segment $a_1 = \{y_1, \$\}$) and x_4 waking (emitting segment $a_4 = \{y_2, y_3, \$\}$) while x_2, x_3 and x_5 sleeping (emitting empty segment, i.e., $a_2 = \{\$\}$, $a_3 = \{\$\}$, and $a_5 = \{\$\}$).

of the probabilities of all the segmentations in $\mathcal{S}_y$,

$$p(y_{1:T}|x_{1:T'}) \triangleq \sum_{a_{1:T'} \in S_y} \prod_{t=1}^{T'} p(a_t|x_t),$$

$$\mathcal{S}_y \triangleq \{a_{1:T'} : \pi(a_{1:T'}) = y_{1:T}\}.$$

where the $p(a_t|x_t)$ is the segment probability given input element x_t, which is usually modeled using a recurrent neural network (RNN) with a softmax probability function. $\pi(\cdot)$ is the concatenation operator and the symbol \$ will be ignored in the concatenation operator $\pi(\cdot)$. Since $|S_y|$ is exponentially large, direct summation quickly becomes infeasible when T or T' is not small. Instead, Wang et al. (2017) developed an exact dynamic programming algorithm to tackle the computation. The authors also discussed ways to carry over information across segments using a separate RNN, which we will not elaborate here.

SWAN defines a conditional probability for an output sequence given an input one. It can be used in many sequence-to-sequence tasks. In practice, a sequence encoder like a bidirectional RNN can be used to process the raw input sequence (like speech signals or source language) to obtain $x_{1:T'}$ that is to be passed into SWAN.

2.2. Local reordering of input sequences

SWAN assumes a monotonic alignment between the output segments and the input elements. For speech recognition experiments in Wang et al. (2017), this is a reasonable assumption. However for machine translation, this might be too strict. In neural machine translation literatures, attention mechanisms were proposed to address alignment problems (Bahdanau et al., 2014; Luong et al., 2015; Raffel et al., 2017). But it is not clear how to apply a similar attention mechanism to SWAN due to the segmentations of the output sequences.

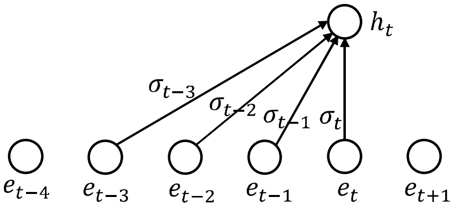

Figure 2. Example of a local reordering layer of window size $\tau = 4$ to compute h_t. Here $\sigma_{t-4+i} \triangleq \sigma(w_i^T[e_{t-3}; e_{t-2}; e_{t-1}; e_t])$, $i = 1, 2, 3, 4$, are the gates that decides how much information h_t should accept from those elements from this input window.

We first note that in using SWAN, a bidirectional RNN encoder for the source language can partly mitigate the alignment issue, since it can access every source word. However, we found it is not enough to obtain superior performance. Here, we propose a new reordering layer that does (soft) local reordering of the input sequence. Together with SWAN, we obtain better performances on the IWSLT 2014 German-English translation task. One additional advantage of not using attention model is that the decoding can be much faster, removing the need to query the entire input source for every output word (Raffel et al., 2017).

We now describe the details of the local reordering layer. Let the input to the local reordering layer be $e_{1:T'}$ and the output of this layer is $h_{1:T'}$. We compute h_t as

$$h_t = \tanh\left(\sum_{i=1}^{\tau} \sigma\left(w_i^T[e_{t-\tau+1}; ...; e_t]\right) e_{t-\tau+i}\right).$$

where $\sigma(\cdot)$ is the sigmoid function and τ is the local reordering window size. Notation $[e_{t-\tau+1}; ...; e_t]$ is the concatenation of vectors $e_{t-\tau+1}, e_{t-\tau}, ..., e_t$. For $i = 1, ...\tau$, notation w_i is the parameter for sigmoid function at position i of the input window. It decides the weight of $e_{t-\tau+i}$ through gate $\sigma\left(w_i^T[e_{t-\tau+1}; ...; e_t]\right)$. The final output h_t is a weighted linear combination of the input elements $e_{t-\tau+1}, e_{t-\tau}, ..., e_t$ in the window followed by a nonlinear transformation by the $\tanh(\cdot)$ function.

Figure 2 illustrates the idea. Here we want to (soft) select an input element from a window given all information available in this window. Suppose we have two adjacent windows, $(e_t, ..., e_{t+\tau-1})$ and $(e_{t+1}, ..., e_{t_\tau})$. If we pick $e_{t+\tau-1}$ in the first window and e_{t+1} in the second, e_{t+1} and $e_{t+\tau-1}$ are effectively reordered as long as τ is larger than 2. We design our layer differently from the typical attention mechanism (Bahdanau et al., 2014) in two ways because we do not have a query to begin with as in standard attention mechanisms. First, we do not normalize the weights for the input elements $e_{t-\tau+1}, e_{t-\tau}, ..., e_t$. This provides the reordering capability and can shut off everything if needed. Second, the weight of any position i in the reordering window is determined by all input elements $e_{t-\tau+1}, e_{t-\tau}, ..., e_t$ in the window.

One other related work to this layer is the Gated Linear Units (GLU) (Dauphin et al., 2016) which can control the

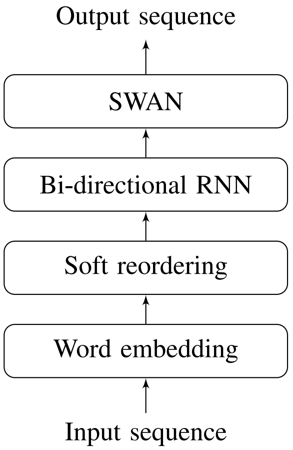

Figure 3. The overall architecture of NPMT.

information flow of the output of a traditional convolutional layer. But GLU does not have the ability to choose which input elements from the convolution window. And in our experiments, we found neither GLU nor traditional convolutional layer helped our setup of using SWAN.

Figure 3 shows the overall architecture of NPMT. In our experiments, we use one local reordering layer and one or two bidirectional RNN layers.

3. Preliminary experiments

In our experiment, we evaluate our model on the German-English machine translation track of the IWSLT 2014 evaluation campaign (Cettolo et al., 2014). The data comes from translated TED talks, and the dataset contains roughly 153K training sentences, 7K development sentences, and 7K test sentences. We use the same preprocessing and dataset splits as in Ranzato et al. (2015); Wiseman & Rush (2016); Bahdanau et al. (2017).

We report our IWSLT 2014 experiments using a two-layer GRU encoder and a two-layer GRU decoder, each with 256 hidden units. We add dropout with a rate of 0.35 in the GRU layer. The maximum segment length is set to 6 and the window size for the reordering layer is 6. Batch size is set as 32 and the Adam algorithm (Kingma & Ba, 2014) is used for optimization with initial learning rate as 0.001. For decoding, we use greedy search and beam search with a beam size of 10. As reported in Maas et al. (2014); Bahdanau et al. (2017), we find that penalizing candidate sentences that are too short was required to obtain the best results. All hyperparameters are chosen based on the development set. For the baseline sequence-to-sequence model with the log-likelihood objective, the best result is obtained with one-layer encoder and one-layer decoder.[2]

	BLEU	
	Greedy	Beam Search
MIXER (Ranzato et al., 2015)	20.73	21.83
LL (Wiseman & Rush, 2016)	22.53	23.87
BSO (Wiseman & Rush, 2016)	23.83	25.48
LL (Bahdanau et al., 2017)	25.82	27.56
RF-C+LL (Bahdanau et al., 2017)	27.70	28.30
AC+LL (Bahdanau et al., 2017)	27.49	28.53
NPMT (this paper)	**27.83**	**28.96**
NPMT+LM (this paper)	–	**29.16**

Table 1. Translation results on the test set. MIXER (Ranzato et al., 2015) uses a convolutional encoder and simpler attention. LL (attention model with log likelihood) and BSO (beam search optimization) of Wiseman & Rush (2016), and LL, RF-C+LL, and AC+LL of Bahdanau et al. (2017) use a one-layer GRU encoder and decoder with attention. (RF-C+LL and AC+LL are different settings of actor-critic algorithms combined with LL.)

We also explore an option of adding a language-model score during beam search as the traditional statistical machine translation does. This option does not make much sense in attention-based approaches, since the decoder itself is usually a neural network language model. In SWAN, however, there is no language models directly involved in the segmentation modeling[3] and we find it useful to have an external language model during beam search. We use a 3-order language model trained using KenLM implementation (Heafield et al., 2013) for English target training data. So the final beam search score we use is

$$Q(y) = \log p(y|x) + \lambda_1 \text{word\_count}(y) + \lambda_2 \log p_{\text{lm}}(y),$$

where we empirically find that $\lambda_1 = 1.2$ and $\lambda_2 = 0.2$ gives good performance. If no external language models are used, we set $\lambda_2 = 0$. This scoring function is similar to the one for speech recognition in Hannun et al. (2014).

The results are summarized in Table 1. NPMT achieves state-of-the-art results on this dataset as far as we know. Compared to the supervised sequence-to-sequence model, LL (Bahdanau et al., 2017), NPMT achieves 2.01 BLEU gain in the greedy setting and 1.4 BLEU gain using beam-search. Our results are also better than those from the actor-critic based methods in Bahdanau et al. (2017). But we note that our proposed method is orthogonal to the actor-critic

[2] We explored several settings of different number of layers and dropout options but did not find better results than the one reported in Bahdanau et al. (2017).

[3] In Wang et al. (2017), SWAN does have an option to use a separate RNN that connects the segments, which can be seen as a language model. However, different from speech recognition experiments, we find in machine translation experiments, adding this separate RNN leads to a worse performance. We suspect this is because that a RNN language model can be easier to learn than the segmentation structures and SWAN gets stuck in that local mode. This is further evidenced by the fact that the average segment length is much shorter with a separate RNN in SWAN.

Table 2. Examples of German-English translation outputs with their segmentations, where "•" represents the segment boundary.

source	danke , aber das beste kommt noch .
target ground truth	thanks . i haven 't come to the best part .
greedy decoding	thank you • , • but • the best thing • is still coming • .
source	sie können einen schalter dazwischen einfügen und so haben sie einen kleinen UNK erstellt .
target ground truth	you can put a knob in between and now you 've made a little UNK .
greedy decoding	you can put • a • switch • in between • , and • so • they made • a little • UNK • .
source	sie wollen die entscheidung wirklich richtig treffen , wenn es für alle ewigkeit ist , richtig ?
target ground truth	you really want to get the decision right if it 's for all eternity , right ?
greedy decoding	you really want to make • the decision • right • if • it 's • for • all • eternity • , • right • ?
source	es gibt zehntausende maschinen rund um die welt die kleine stücke von dna herstellen können , 30 bis 50 buchstaben lang aber es ist ein UNK prozess , also je länger man ein stück macht , umso mehr fehler passieren .
target ground truth	there are tens of thousands of machines around the world that make small pieces of dna – 30 to 50 letters - in length - and it 's a UNK process , so the longer you make the piece , the more errors there are .
greedy decoding	there are • tens of thousands of • machines • around • the world • can make • the little • pieces • of • dna • , • 30 • to • 50 • letters • long • , but • it 's • a • UNK • process • , • so • the • longer • you do • a • piece • , • the • more • mistakes • .

method. So it is possible to further improve our results using the actor-critic method. Finally, with a language model added during beam search, NPMT+LM achieves an even higher BLEU score of 29.16.

We also run two following experiments to verify the sources of the gain. The first is to add a reordering layer to the original sequence-to-sequence model with attention, which gives us BLEU scores of 25.55 (greedy) and 26.91 (beam search). The second is to remove the reordering layer from NPMT, which gives us BLEU scores of 25.47 (greedy) and 27.05 (beam search). This shows that the reordering layer and SWAN are both vital for the effectiveness of NPMT.

In greedy decoding, we can estimate the *average segment length*[4] for the output. The average segment length is around 1.3–1.4, indicating some phrases are being decoded. Table 2 shows some randomly sampled examples. We can observe there are many informative segments in the decoding results, e.g., "tens of thousands of", "the best thing", "a little", etc.

4. Conclusion

We studied neural phrase-based machine translation using SWAN, a segmentation-based sequence modeling technique. We also introduced a local reordering layer to alleviate the monotonic alignment requirement in SWAN. Our preliminary experimental results showed promising results on a German-English translation task. We plan to explore larger datasets and more language pairs in future work.

References

Bahdanau, Dzmitry, Cho, Kyunghyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Bahdanau, Dzmitry, Brakel, Philemon, Xu, Kelvin, Goyal, Anirudh, Lowe, Ryan, Pineau, Joelle, Courville, Aaron C., and Bengio, Yoshua. An actor-critic algorithm for sequence prediction. In *International Conference on Learning Representations*, 2017.

Cettolo, Mauro, Niehues, Jan, Stüker, Sebastian, Bentivogli, Luisa, and Federico, Marcello. Report on the 11th IWSLT evaluation campaign, IWSLT 2014. In *Proceedings of IWSLT*, 2014.

Dauphin, Yann N, Fan, Angela, Auli, Michael, and Grangier, David. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*, 2016.

Hannun, Awni, Case, Carl, Casper, Jared, Catanzaro, Bryan, Diamos, Greg, Elsen, Erich, Prenger, Ryan, Satheesh, Sanjeev, Sengupta, Shubho, Coates, Adam, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

Heafield, Kenneth, Pouzyrevsky, Ivan, Clark, Jonathan H., and Koehn, Philipp. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pp. 690–696, 2013.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koehn, Philipp. *Statistical Machine Translation*. Cambridge University Press, 2009.

Koehn, Philipp, Och, Franz Josef, and Marcu, Daniel. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pp. 48–54. Association for Computational Linguistics, 2003.

[4] The average segment length is defined as the length of the output (excluding end of segment symbol $) divided by the number of segments (not counting the ones only containing $).

Lopez, Adam. Statistical machine translation. *ACM Computing Surveys (CSUR)*, 40(3):8, 2008.

Luong, Minh-Thang, Pham, Hieu, and Manning, Christopher D. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Maas, Andrew L., Hannun, Awni Y., Jurafsky, Daniel, and Ng, Andrew Y. First-pass large vocabulary continuous speech recognition using bi-directional recurrent DNNs. *CoRR*, abs/1408.2873, 2014.

Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Raffel, Colin, Luong, Thang, Liu, Peter J, Weiss, Ron J, and Eck, Douglas. Online and linear-time attention by enforcing monotonic alignments. In *International Conference on Machine Learning (ICML)*, 2017.

Ranzato, Marc'Aurelio, Chopra, Sumit, Auli, Michael, and Zaremba, Wojciech. Sequence level training with recurrent neural networks. *CoRR*, abs/1511.06732, 2015.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

Wang, Chong, Wang, Yining, Huang, Po-Sen, Mohamed, Abdelrahman, Zhou, Dengyong, and Deng, Li. Sequence modeling via segmentations. In *International Conference on Machine Learning (ICML)*, 2017.

Wiseman, Sam and Rush, Alexander M. Sequence-to-sequence learning as beam-search optimization. *CoRR*, abs/1606.02960, 2016.