Word-Level Error Correction in Non-autoregressive Neural Machine Translation

Ziyue Guo, Hongxu Hou(✉), Nier Wu, and Shuo Sun

College of Computer Science-college of Software, Inner Mongolia University,
Hohhot, China
guoziyue08@126.com, cshhx@imu.edu.cn, wunier04@126.com, sunshuo07@126.com

Abstract. Non-Autoregressive neural machine translation (NAT) not only achieves rapid training but also actualizes fast decoding. However, the implementation of parallel decoding is at the expense of quality. Due to the increase of speed, the dependence on the context of the target side is discarded which resulting in the loss of the translation contextual position perception ability. In this paper, we improve the model by adding capsule network layers to extract positional information more effectively and comprehensively, that is, relying on vector neurons to compensate for the defects of traditional scalar neurons to store the position information of a single segment. Besides, word-level error correction on the output of NAT model is used to optimize generated translation. Experiments show that our model is superior to the previous model, with a BLEU score of 26.12 on the WMT2014 En-De task and a BLEU score of 31.93 on the WMT16 Ro-En, and the speed is even more than six times faster than the autoregressive model.

Keywords: Non-autoregressive neural machine translation · Word-level error correction · Capsule network

1 Introduction

Most neural machine translation (NMT) [1,2] models are sequentially autoregressive models (AT) such as RNNs, Transformer [3] which have state-of-the-art performance. The training process of Transformer is parallel, but in decoding phase, it exploit the generated sequence to predict the current target word which will cause severe decoding delay. In recent years, non-autoregressive neural machine translation model (NAT) [4] is proposed to effectively speed up the decoding process which exploits Knowledge Distillation [5] and fine-tuning to assist training. Subsequently, there are some novel-innovative improvements based on the NAT model, such as the work of regulating the similarity of hidden layer states by two auxiliary regularization terms [6], the model reconstruct generative translation through iterative refinement [7] and Ghazvininejad put forward to partially mask target translation through the conditional masked language model [8].

H. Yang et al. (Eds.): ICONIP 2020, CCIS 1332, pp. 726–733, 2020.
https://doi.org/10.1007/978-3-030-63820-7_83

In this work, we propose to utilize the Capsule Network [9] in the architecture which has a significant impact on extracting more deeply positional features and making the generated translation more advantageous in word order. Besides, we adopt the word-level error correction method to reconstruct the generated sentence which can alleviate the translation problems. Experiments show that our model is superior to the previous NAT models. On the WMT14 De-En task, the addition of the capsule network layers increases the BLEU score by more than 6. More significantly, our word-level error correction method brings 1.88 BLEU scores improvement. We also perform case study on WMT14 En-De and ablation study on IWSLT16 to verify the effectiveness of the proposed methods.

2 Background

2.1 Non-autoregressive Neural Machine Translation

Under the condition of given source sentence $S = (s_1, ..., s_K)$ and target sentence $T = (t_1, ..., t_L)$, the autoregressive model utilizes a sequential manner to predict the current word which will bring a certain degree of delay. Non-autoregressive neural machine translation model (NAT) [4] is proposed to improve the decoding speed which only predicts based on the source sequence and the target sequence length L_y predicted in advance:

$$P_{NAT}(T|S;\theta) = P(L_y|S;\theta) \cdot \prod_{l}^{L_y} P(t_l|S;\theta) \tag{1}$$

where θ is a series of model parameters.

2.2 Neural Machine Translation with Error Detection

For error detection in NMT, the model first characterizes each word in the source sentence as a word embedding vector and then feeds it to the bidirectional LSTM. At each time step, the hidden state in both directions is combined and regarded as the final output. In addition, the error correction model also constructs mismatching features, that is when there are wrong words in a output sequence, the pre-trained model will give the correct word prediction distribution and there will be a gap between their probability distributions. The model make the next prediction according to this gap feature, as shown in Eq. 2.

$$argmin \sum_{k=1}^{T} XENT\left(g_k, W\left[\overrightarrow{h_k}, \overleftarrow{h_k}, \overrightarrow{h_{k+1}}, \overleftarrow{h_{k+1}}\right]\right) \tag{2}$$

where $XENT$ stands for cross-entropy loss, W represents the weight matrix, $\overrightarrow{h_k}, \overleftarrow{h_k}$ means the overall score of the sentence in the forward and backward directions and g_k is the gap label between k-th token and k+1st token.

3 Approach

3.1 Model Architecture

Since the NAT model ignores the target words and context information, we use the Capsule Network [9] to improve, the model architecture is shown in Fig. 1 which also composed of encoder and decoder. The hidden layer state of the encoder is shown in the Eq. 2.

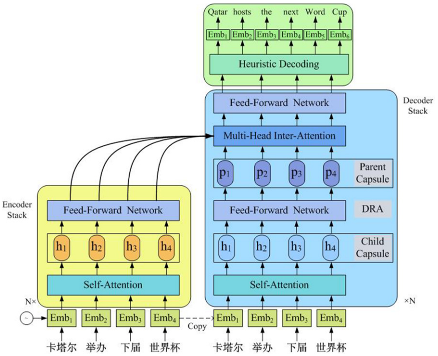

Fig. 1. The architecture of the proposed NAT-CN model. The encoder use child layer to capture location information and the decoder integrate information by parent layer, then update the weights by Dynamic Routing Algorithm (DRA).

$$h_j = \sum_i \alpha_{ij} F(e_i, w_{ij}) \tag{3}$$

where e_i is the output of the self-attention layer, α represents the coupling coefficient of the capsule network, and the final output of this layer is h_j.

Similar to the encoder side, we use a child layer to extract source information, but at decoder side we introduce an additional parent layer to integrate information extracted by the previous layer (ie, child layer), and map it to another form that is consistent with the parent's representation:

$$s_j = \sum_i^M F(h_{ij}, w_{ij}) \tag{4}$$

where M represents the number of child capsules in the child capsule layer. Then use the Squashing function to compress the modulus of the vector into the interval [0, 1), each parent capsule will update the state as follows:

$$p_j = Squash(s_j) = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{5}$$

Integrate all the child capsules in the form described above to generate the final parent capsule layer representation $P = [p_1, p_2, ..., p_N]$. After that, iterative updating is used to determine what information in the N parent capsules will be transmitted to the Multi-Head Inter-Attention sub-layer.

$$Attention(Q_p, K_p, V_p) = softmax(\frac{Q_p K_p^T}{\sqrt{d_k}}) \cdot V_p \tag{6}$$

where Q_p is the output of the parent capsule layer, K_p, V_p are vectors from the encoder, and they all contain rich position information.

Position-Aware Strategy. Since there is no direct target sequence information on the decoder side, we combine the extracted deeper information with the source information to get the final word vector representation and feed it to the next layer:

$$Emb_p(Q_p, K_p, V_p) = (e_1 + p_1, ..., e_n + p_n) \tag{7}$$

where e_i represents the original source word embedding and p_i indicates the position vector extracted by the capsule network layers. Besides, to accomplish parallel decoding and advantage the decoder to infer, we calculate the ratio λ between target and source sentence lengths in the training set and given a bias term C. The target sentence length +*96 $L_y = \lambda L_x + C$, then predict it from $[\lambda L_x - B, \lambda L_x + B]$, where B represents half of the searching window.

3.2 Training

Objective Function. We utilize teacher model to guide the training of NAT model to improve translation quality. In the capsule network layers, we update the parameters through an iterative dynamic routing algorithm:$b_{ij} = b_{ij} + p_j F(u_i, w_j)$, where u_i is the previous capsule network output, p_j is the parent capsule network layer output and $F(\cdot)$ denotes the calculation of the feed-forward neural network. We use cross-entropy to calculate the loss of NAT model with position awareness during the training phase, as shown in Eq. 8.

$$L_{NAT}(S;\theta) = -\sum_{l=1}^{L_y}\sum_{t_l}((logP_{NAT}(t_l|L_y, S) \cdot logP_{AT}(t_l|t_1, .., t_{l-1}, S;\theta)) \tag{8}$$

We utilize the Sequence-Level Interpolation Knowledge Distillation method [5] to assist training which makes the proposed NAT-CN model generate translations by selecting the output that is closest to the gold reference r but has the highest probability under the guidance of distilled data. The training process is shown in Eq. 9.

$$L_{IKD} = (1-\alpha)L_{SEQ-NLL} + \alpha L_{SEQ-KD} = -(1-\alpha)logp(r|s) - \alpha logp(\hat{t}|s) \tag{9}$$

where α is a hyper-parameter and $\hat{t}$ is the output under the guidance of teacher model.

3.3 Word-Level Error Correction

Teacher Model. For the translation problem of the NAT model, we perform word-level error correction on the generated translation by use bilingual teacher model. As shown in Fig. 1, teacher model extracts features bidirectionally from source sequences and generates the latent variable $\overleftarrow{Z}$ and $\overrightarrow{Z}$, then integrates encoded potential variables to predict the probability distribution of candidate words as the gap feature. We use this gap to guide error correction and obtain the output of teacher model by maximizing the expected probability.

$$p(t|z) = \prod_{l} p(t_l|\overleftarrow{z_l}, \overrightarrow{z_l}); q(z|t,s) = \prod_{l} q(\overleftarrow{z_l}|s, t_{<l}, \overrightarrow{z_l}|s, t_{>l}) \tag{10}$$

where z denotes latent variable, we only need to construct two probabilities of $p(\cdot)$ and $q(\cdot)$ by bidirectional transformer to get the maximum expectation.

Force Decoding. We can extract three kinds of matching features after training teacher model, which consists of latent variable z_l, token embedding E_p and categorical distribution $p(t_k|\cdot) \sim Categorical(softmax(I_k))$. Therefore, we can construct 4-dimensional mis-matching feature $f_k^{mis-match}$:

$$f_k^{mis-match} = (I_{k,m_k}, I_{k,i^k_{max}}, I_{k,m_k} - I_{k,i^k_{max}}, \Xi_{m_k \neq i_{max}}) \tag{11}$$

where m_k represents $k-th$ token in the NAT-CN model output, $i^k_{max} = argmax_i I_k$ is the gap feature. These four items respectively represent:the probability of forced decoding into the current output token m_k; the model does not use forced decoding but retains the probability information of the most likely word i^k_{max}; the difference between the first two items; the probability distribution used to indicate whether the current word is consistent with the predicted word.

Then we can use f_k to forcibly decode the current token into the token with highest probability. We modified the original NAT objective to get Eq. 12.

$$P_{NAT}(T|S, f_k; \theta) = P(L_y|S; \theta) \cdot \prod_{l=1}^{L_y} P(t_l|S, Z, f_k; \theta) \tag{12}$$

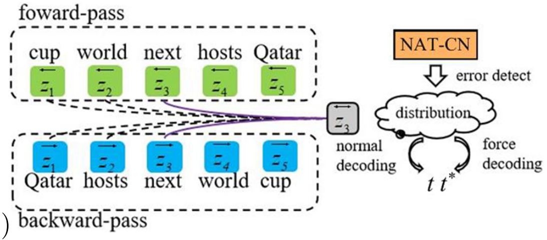
foward-pass
cup world next hosts Qatar
$\overleftarrow{z_1}$ $\overleftarrow{z_2}$ $\overleftarrow{z_3}$ $\overleftarrow{z_4}$ $\overleftarrow{z_5}$
$\overleftrightarrow{z_3}$
$\overrightarrow{z_1}$ $\overrightarrow{z_2}$ $\overrightarrow{z_3}$ $\overrightarrow{z_4}$ $\overrightarrow{z_5}$
Qatar hosts next world cup
backward-pass
NAT-CN
error detect
distribution
normal decoding
force decoding
t t*

Fig. 2. Use the output of the asynchronous bidirectional decoding model to perform word-level error correction on the translation of NAT-CN model.

As shown in Fig. 2, according to this mismatching feature, it can be decided whether the translation of the NAT-CN model is normally decoded to t or forcedly decoded to the reference translation t^*.

4 Experiments and Results

4.1 Datasets and Setting

We use the following three machine translation tasks: WMT14 En-De (4.5M pairs) and WMT16 En-Ro (610k pairs), IWSLT16 En-De (196k pairs). For WMT16, we utilize newsdev2016 as the verification set and newstest2016 as the test set. For IWSLT16, we employe test2013 as development set. For WMT14, we utilize newstest2013 and newstest2014 as the validation set and test set respectively. All datasets are tokenized by Moses[1] and segmented into sub-word units

[1] https://github.com/moses-smt/mosesdecoder.

Table 1. Evaluation of translation quality on select translation tasks including BLEU scores, decoding latency and training speed. Where “NAT-CN” represents the proposed model with capsule network and “EC” refers to the NAT-CN model combined with word-level translation error correction method. We use “KD” to empress the method of knowledge distillation and “i_{dec}” stands for the number of iterations.

Models	WMT14		WMT16		IWSLT16	Latency	Speedup
	En-De	De-En	En-Ro	Ro-En	En-De		
Transformer	27.41	31.29	33.12	33.86	30.90	607 ms	1.00×
NAT-FT	17.69	21.47	27.29	29.06	26.52	39 ms	15.6×
NAT-FT(+NPD $s = 10$)	18.66	22.41	29.02	30.76	27.44	79 ms	7.68×
NAT-IR($i_{dec} = 10$)	21.61	25.48	29.32	30.19	27.11	404 ms	1.5×
NAT-IR($adaptive\ refinements$)	21.54	25.43	29.66	30.30	27.01	–	–
NAT-LV	25.10	–	–	–	–	89 ms	6.8×
FlowSeq-base(+KD)	21.45	26.16	29.34	30.44	–	–	–
FlowSeq-large(+KD)	23.72	28.39	29.73	30.72	–	–	–
CMLM-small($i_{dec} = 4$)	24.17	28.55	30.00	30.43	–	–	–
NAT-REG($rescoring$ 9)	24.61	28.90	–	–	27.02	40 ms	15.1×
NAT-CN(B = 0,1 candidates)	23.10	25.25	28.50	29.87	26.59	45 ms	13.47×
NAT-CN(B = 4,9 candidates)	24.92	27.47	29.69	30.31	27.05	72 ms	8.43×
NAT-CN(+EC, B = 4)	26.12	29.35	30.26	31.93	27.79	98 ms	6.18×

by BPE algorithm. We compare our model with strong baseline systems, including the NAT with fertility and noisy parallel decoding (NAT-FT+NPD) [4][2] and our model is modified on it, the NAT with iterative refinement (NAT-IR) [7], the NAT with discrete latent variables (NAT-LV) [11], the conditional sequence generation model with generative flow (FlowSep) [12], the Mask-Predict model (CMLM) [8] and the NAT with auxiliary regularization (NAR-REG) [6].

On the dataset WMT, our parameter settings are the same as Transformer [3] which are described in its paper. Because IWSLT is smaller, the word vector dimension set to 278, the number of hidden layer neurons set to 507, layer depth set to 5, and the attention head set to 2. We conduct experimental verification on the development set and finally select 0.6 as hyper-parameter α in Eq. 9 and the number of parent capsules N and child capsules M are both set to 6. Latency is calculated as the average decoding time of each sentence on entire test set without mini-batching and we test it on two NIVDIA TITAN X.

4.2 Analysis

Results. The experimental results are shown in Table 1. Specifically, on the WMT En→De task, our NAT-CN model get 24.92 BLEU[3] scores, which is an improvement of 6.26 BLEU scores compared to the NAT-FT(+NPD) model. After combining the word-level error correction method, we get 26.12 BLEU scores which is an improvement of 1.02 compared with the best baseline NAT-LV model and has a similar decoding speed, however, the difference is only

[2] Source code of this work is available at https://github.com/salesforce/nonauto-nmt.

[3] https://github.com/harpribot/nlp-metrics/tree/master/bleu.

Table 2. Translation case studies on WMT14 De→En task. In order to compare under the same conditions, we set B to 4 in the experiment.

Source	im jahr 2000 wurden weltweit etwa 100 milliarden fotos geschossen, aber nur ein winziger teil davon wurde ins netz geladen.
Reference	around 100 billion photographs were taken worldwide in 2000,but only a tiny part of them was uploaded.
AT	around 100 billion photos were taken worldwide in 2000, but only tiny part of them was uploaded.
NAT-FT	taken worldwide in 2000 about 100 billion photos photos , but uploaded only little part of them was was.
NAT-CN	about 100 billion photos photos taken worldwide in 2000, but only little part of them was was uploaded.
NAT-CN(+EC)	around 100 billion photos were taken worldwide in 2000, but only little part of them was [null] uploaded.

1.29 compared with the Transformer but the decoding speed is improved by 6.18 times. On the En-Ro task, the BLEU scores of 30.26 and 31.93 are finally obtained, and the word-level error correction method on Ro→En also brings 1.62 BLEU scores improvement.

Case Study and Ablation Study. A translation case on WMT14 De-En is shown in Table 2. We utilize Transformer [3] as AT model and set B to 4. Compared with the original NAT-FT model [4], our NAT-CN model has a better ability to capture the global position information, and the effect of the word-level error correction method is also significant. There is a gap in the word order between the NAT-FT model translation and the reference and there are also translation problems such as "photos photos" and "was was". However, our model corrects "photos" to "were" and "was" to "null", that is the target word at the current position is empty, and also corrects "about" to "around". We mark the corrected words in red font.

Table 3. Ablation study performance on IWSLT16 development set.

Model variants	BLEU	Latency	Speedup
NAT-BASE	21.69	36 ms	16.86×
NAT-BASE(+CN)	25.61	59 ms	10.28×
NAT-BASE(+EC)	28.24	74 ms	8.20×
NAT-BASE(+Both)	28.81	93 ms	7.31×

We perform ablation study on the IWSLT16 translation task to verify the impact of different methods. As shown in Table 3, after using the capsule network layers, the BLEU score of our model is increased by about 4 and the decoding speed also improved by 16.86 times. It is enough to see the impact of the increase of the capsule network layers on the overall experimental results. After combining the word-level error correction method, the BLEU score improves 2.63 which also proves that this approach can make the translation close to the output of the autoregressive model.

5 Conclusion

We propose a novel NAT model architecture to extract the position feature and its context of the word embedding by adding capsule network layers to the vanilla NAT model. In addition, the word-level error correction method is used to reconstruct the translation of the NAT model, which reduces the degradation of the model while improving the decoding speed. Experiments show that our model has a significant effect compared to all non-autoregressive baseline systems.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015
2. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, 8–13 December 2014
3. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017, pp. 5998–6008 (2017)
4. Gu, J., Bradbury, J., Xiong, C., Li, V.O.K., Socher, R.: Non-autoregressive neural machine translation. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018
5. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR (2015). http://arxiv.org/abs/1503.02531
6. Wang, Y., Tian, F., He, D., Qin, T., Zhai, C., Liu, T.: Non-autoregressive machine translation with auxiliary regularization. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, pp. 5377–5384 (2019)
7. Lee, J., Mansimov, E., Cho, K.: Deterministic non-autoregressive neural sequence modeling by iterative refinement. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018, pp. 1173–1182. https://www.aclweb.org/anthology/D18-1149
8. Ghazvininejad, M., Levy, O., Liu, Y., Zettlemoyer, L.: Mask-predict: parallel decoding of conditional masked language models. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP (2019)
9. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, pp. 3856–3866 (2017)
10. Zhou, L., Zhang, J., Zong, C.: Synchronous bidirectional neural machine translation. TACL **7**, 91–105 (2019)
11. Shu, R., Lee, J., Nakayama, H.: Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. CoRR (2019)
12. Ma, X., Zhou, C., Li, X., Neubig, G.: Flowseq: non-autoregressive conditional sequence generation with generative flow. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP (2019)