A Multi-Hop Attention for RNN based Neural Machine Translation

Shohei Iida[†], **Ryuichiro Kimura**[†], **Hongyi Cui**[†], **Po-Hsuan Hung**[†],
Takehito Utsuro[†] **and Masaaki Nagata**[‡]
[†]Graduate School of Systems and Information Engineering, University of Tsukuba, Japan
[‡]NTT Communication Science Laboratories, NTT Corporation, Japan

Abstract

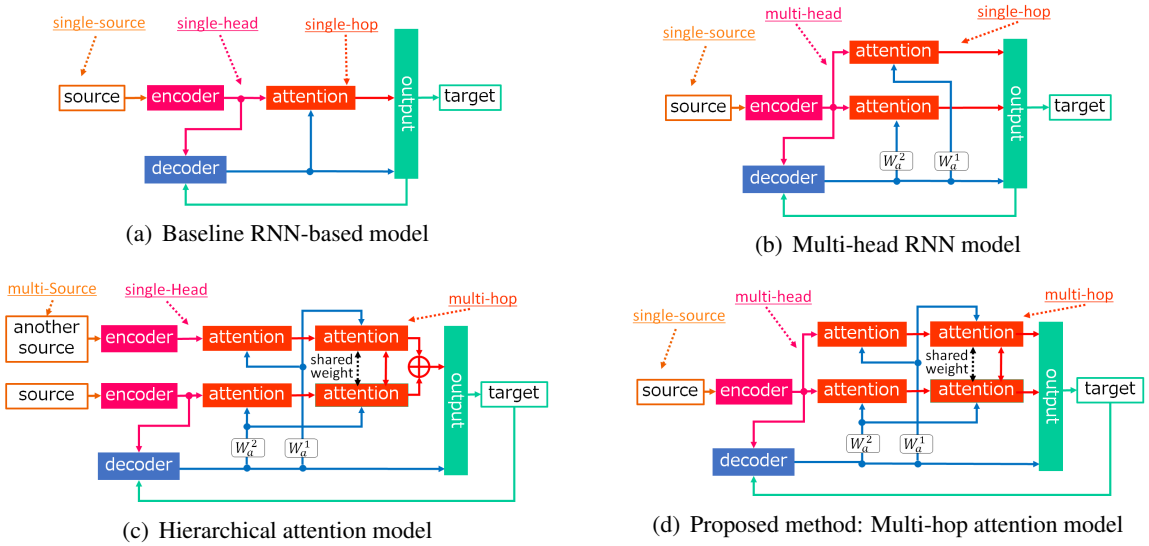
Among recent progresses of neural machine translation models, the invention of the Transformer model is one of the most important progresses. It is well-known that the key technologies of the Transformer include multi-head attention mechanism. This paper introduces the multi-head attention mechanism into the traditional RNN-based neural machine translation model. Moreover, inspired by the existing multi-hop architectures such as end-to-end memory networks and convolutional sequence to sequence learning model, this paper proposes an RNN based NMT model with a multi-hop attention mechanism. The proposed multi-hop attention model has two heads, where for each head, a context vector is calculated based on the states of the encoder and the decoder. Then, in the second turn of the context vector calculation, those context vectors are updated depending not only on one's own context vector but also on the context vector of the other head. Experimental results show that the proposed model significantly outperforms the baseline in BLEU score in Japanese-to-English/English-to-Japanese machine translation tasks with and without extended context.

1 Introduction

RNN encoder-decoder model (Bahdanau et al., 2015; Luong et al., 2015; Sutskever et al., 2014) was the state-of-the-art in machine translation. However, it is outperformed by non-recursive encoder-decoder models such as Transformer (Vaswani et al., 2017) and Convolutional Sequence-to-Sequence (Gehring et al., 2017) in recent years. However, RNN is not considered to be inferior to Transformer in all respects. For example, according to Tran et al. (2018), it is reported that Transformer is not good at decoding sentences whose length is not included in the training data and it is weak to long distance dependency. In other words, it is weak against long sentence translation. It seems that Transformer became more powerful than RNN by increasing the number of parameters, but it became weak to long sentences for the same reason.

We propose an RNN based source-to-target attention mechanism where the number of parameters increases by repeating the calculation of multi-head attention for a single-source encoder like multi-hop attention in end-to-end memory networks (Sukhbaatar et al., 2015). In the proposed mechanism, those increased number of parameters are well-tuned so that the overall translation accuracy improves, in particular, for long sentences. The proposed multi-hop attention mechanism is based on the hierarchical attention (Libovický and Helcl, 2017) for multi-source encoders, although, in the hierarchical attention (Libovický and Helcl, 2017), the number of parameters for one input does not increase, unlike in the proposed multi-hop attention mechanism.

In evaluation, we compared the performance of the proposed method with Transformer and RNN encoder-decoder using OpenSubtitles 2018 (Lison et al., 2018) and Asian Scientific Paper Excerpt Corpus (ASPEC) (Nakazawa et al., 2016). To test the power of translating long sentences, we also



(a) Baseline RNN-based model

(b) Multi-head RNN model

(c) Hierarchical attention model

(d) Proposed method: Multi-hop attention model

Figure 1: Baseline attention and proposed attention

made a context-aware translation model, called 2-to-2 (Bawden et al., 2018; Tiedemann and Scherrer, 2017) for OpenSubtitles 2018. In the Japanese-to-English translation of the ASPEC corpus, the proposed method achieved a significantly better score than the Transformer for long sentences with more than 120 tokens.

In the following sections, we first show previous works on baseline RNN and multi-head RNN encoder-decoders in Section 2. We then describe the proposed multi-hop method in Section 3. We then show the performance for Japanese-to-English and English-to-Japanese translation tasks, focusing on long sentences in Section 4.

2 Neural Machine Translation

2.1 RNN based sequence to sequence NMT

There are two distinctive features in sequence-to-sequence model (Bahdanau et al., 2015; Luong et al., 2015) using RNN (Figure 1(a)). One point is that its encoder and decoder can naturally handle time series and the other point is that it can decide which encoder states in the time series the decoder should pay attention to by introducing a mechanism called source-target attention (Bahdanau et al., 2015; Luong et al., 2015).

In other words, the source-target attention of RNN is designed to deal with time series compared with the self-attention of Transformer where time series are artificially represented using positional embeddings (Vaswani et al., 2017). In this paper, considering this point, we propose a novel model suitable for long sentences by efficiently increasing the number of parameters for source-target attention.

2.2 Multi-head Attention

In this paper, we define multi-head attention with N heads as follows, where $k\ (=1,\ldots,N)$ denotes the index of the k-th head and $i\ (=1,\ldots,I)$ denotes the index of the i-th word.

$$s_i^{(k)} = W_a^{(k)} d_i \quad (1)$$

$$c_i^{(k)} = softmax(s_i^{(k)} H^{\mathrm{T}}) H \quad (2)$$

In equation (1), the output of RNN decoder d_i is duplicated and converted differently with the weights into multi-head. $W_a^{(k)}$ is a learnable parameter, which duplicated and converted d_i to $s_i^{(k)}$.

In equation (2), dot product attention (Luong et al., 2015; Vaswani et al., 2017) is used to calculate the context vector $c_i^{(k)}$ between k-th head of a decoder state $s_i^{(k)}$ and encoder states H.

When the model has two heads ($N = 2$), the equation (1) and the equation (2) becomes as follows.

$$s_i^{(1)} = W_a^{(1)} d_i \quad (3)$$

$$s_i^{(2)} = W_a^{(2)} d_i \quad (4)$$

$$c_i^{(1)} = softmax(s_i^{(1)} H^{\mathrm{T}}) H \quad (5)$$

$$c_i^{(2)} = softmax(s_i^{(2)} H^{\mathrm{T}}) H \quad (6)$$

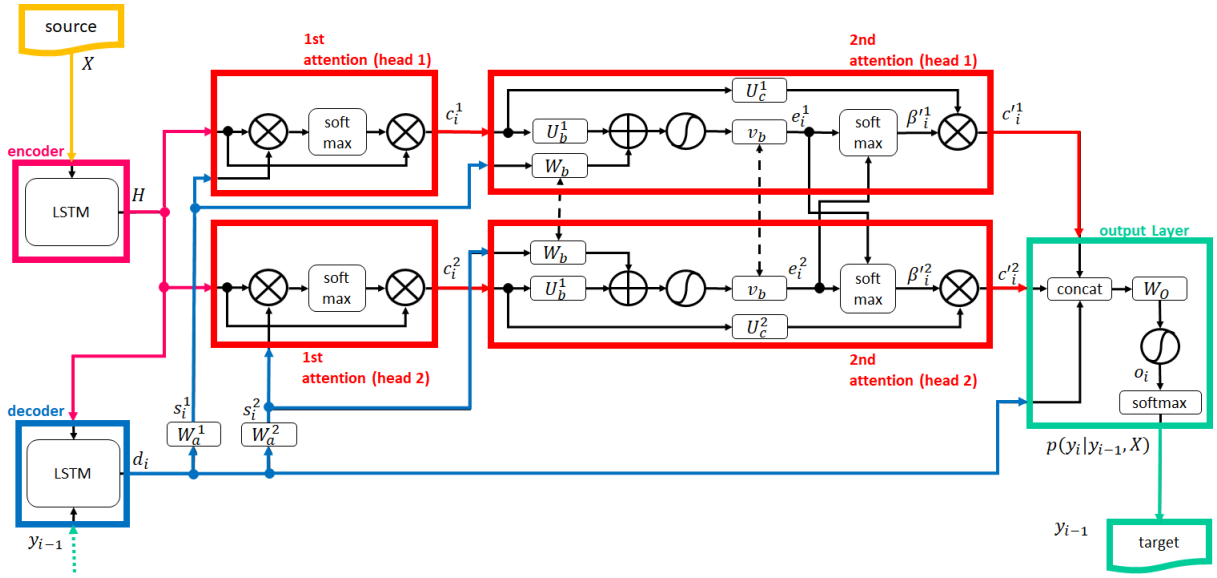


Figure 2: Proposed method detail

As shown in the equation (5) and the equation (6), by using multiple parallel attention via the parameters $W_a^{(k)}$, we expect that each head will attend to a different part of the encoder states.

Chen et al. (2018) attempted to incorporate the various mechanisms of the Transformer into RNN encoder-decoder. They used multi-head attention as shown in Figure 1(b) in source-target attention. Our method becomes the same as their method when we use single-hop attention.

3 Multi-Hop Attention RNN

3.1 Multi-Hop Dependent Attention

To the best of our knowledge, multi-hop attention is first used in end-to-end memory network (Sukhbaatar et al., 2015) to extend the expressive power of RNN. To introduce multi-hop attention into translation, we refer to hierarchical attention (Libovický and Helcl, 2017) in multimodal translation, which combines the context vector obtained from the text and the intermediate expression vector for an image obtained using CNN.

$$e_i^{(k)} = v_b^{\mathrm{T}} tanh(W_b s_i^{(k)} + U_b^{(k)} c_i^{(k)}) \quad (7)$$

$$\beta_i^{(k)} = \frac{exp(e_i^{(k)})}{\sum_{n=1}^{N} exp(e_i^{(n)})} \quad (8)$$

$$c_i'^{(k)} = \beta_i^{(k)} U_c^{(k)} c_i^{(k)} \quad (9)$$

Equation to compute context vector is defined as equation (7), equation (8), and equation (9). Figure 2 is a detailed diagram of the proposed method.

Table 1: Difference between the proposed method and previous studies

Method	source	head	hop
Baseline RNN	single	single	single
Multi-head RNN	single	multi	single
Hierarchical attention	multi	single	multi
Proposed method	single	multi	multi

To illustrate the difference, the proposed method and hierarchical attention are shown in Figure 1(d) and Figure 1(c) and their difference is summarized in Table 1.

In hierarchical attention, since attention is calculated between states of each encoder for multiple source and states of a single decoder, it uses a single-head for each source. On the other hand, our method uses multiple heads for a single source, where attention is directed to different parts of the source sentence and each head influences each other to learn better feature representation. In equation (7), we calculate the attention score between a decoder state $s_i^{(k)}$ and output of the head of the previous hop $c_i^{(k)}$ using Multi Layer Perceptron (MLP) attention (Luong et al., 2015).

The reason for adopting the MLP attention for the second hop instead of the dot product attention used in the first hop (equation (2)) is that the weight of each head can be shared. Since the parameters W_b and v_b in the equation (7) and Figure 2 are shared by all heads, we expect each head can influence each other. According to the report of Vaswani et al. (2017), it is said that dot product attention is superior to MLP attention. However,

since it has no parameters to be shared, we assume it is not suitable as an attention mechanism for the second hop.

The equation (8) normalizes the attention score of each head to $\beta_i^{(k)}$ by softmax where n ranges over all heads[1]. Finally, a new context vector $c_i'^{(k)}$ is calculated by learnable parameter $U_c^{(k)}$, $\beta_i^{(k)}$, and $c_i^{(k)}$.

When the number of heads N is 2, the above calculation procedure becomes the following:

$$e_i^{(1)} = v_b^{\mathrm{T}} tanh(W_b s_i^{(1)} + U_b^{(1)} c_i^{(1)}) \quad (10)$$

$$e_i^{(2)} = v_b^{\mathrm{T}} tanh(W_b s_i^{(2)} + U_b^{(2)} c_i^{(2)}) \quad (11)$$

$$\beta_i^{(1)} = \frac{exp(e_i^{(1)})}{exp(e_i^{(1)}) + exp(e_i^{(2)})} \quad (12)$$

$$\beta_i^{(2)} = \frac{exp(e_i^{(2)})}{exp(e_i^{(1)}) + exp(e_i^{(2)})} \quad (13)$$

$$c_i'^{(1)} = \beta_i^{(1)} U_c^{(1)} c_i^{(1)} \quad (14)$$

$$c_i'^{(2)} = \beta_i^{(2)} U_c^{(2)} c_i^{(2)} \quad (15)$$

Finally, we concatenate the N context vectors $c_i'^{(k)}$ with the RNN decoder state d_i to obtain the prediction of the output word distribution $p(y_i|y_{i-1}, X)$ where W_o is a learnable parameter.

$$o_i = tanh(W_o[d_i; c_i'^{(1)}; ...; c_i'^{(k)}]) \quad (16)$$

$$p(y_i|y_{i-1}, X) = softmax(o_i) \quad (17)$$

When the number of heads N is 2, equation (16) becomes the following:

$$o_i = tanh(W_o[d_i; c_i'^{(1)}; c_i'^{(2)}]) \quad (18)$$

3.2 Multi-Hop Independent Attention

In the multi-hop dependent attention described in the previous subsection, we use the information of other heads and share parameters of MLP attention (W_b and v_b) over all heads (equation (7)) to calculate the secondary context vector $c_i'^{(k)}$ (equation (9)).

We also implemented multi-hop independent attention, where the secondary attention is calculated by feed forward neural networks whose parameter is $U_c^{(k)}$ without using MLP attention. In this method, equation (9) is changed as follows.

$$c_i'^{(k)} = U_c^{(k)} c_i^{(k)} \quad (19)$$

In this method, since there are no parameters to be shared among heads and no scaling parameters such as $\beta_i^{(k)}$ in equation (8), information of other heads are not used in the secondary attention.

4 Evaluation

In order to confirm the usefulness of the proposed method, this section describes experimental evaluation results in Japanese-to-English/English-to-Japanese machine translation tasks with and without extended context. we used BLEU (Papineni et al., 2002) as the evaluation measure.

4.1 Data

We used the Japanese-English parallel corpora obtained from OpenSubtitles 2018 (Lison et al., 2018) and Asian Scientific Paper Excerpt Corpus (ASPEC) (Nakazawa et al., 2016).

In OpenSubtitles 2018, the total 2,083,576 parallel sentences are divided into 90.0% training data (1,872,077 sentence pairs), 5% development data (102,724 sentence pairs), and 5% test data (108,775 sentence pairs). OpenSubtitles 2018 is a parallel corpus composed of movie subtitles, and their sentences are ordered along the line of the story of the movie. Therefore, in addition to the data used in machine translation tasks without extended context, we created data for context-aware translation according to Tiedemann and Scherre (2017) as follows.

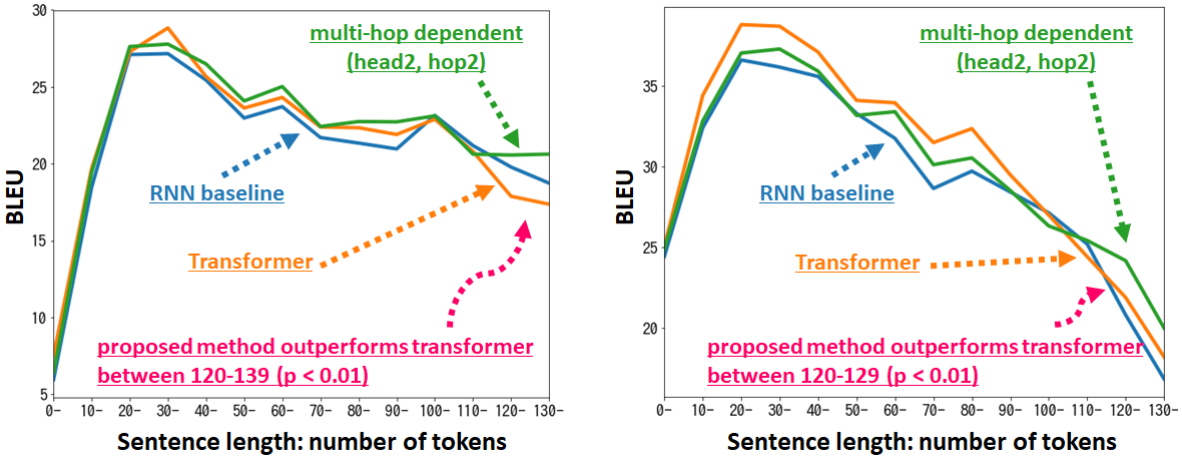
First, given a single pair of a source sentence and a target translated sentence, the source sentence and its immediately preceding sentence are concatenated with a ⟨CONCAT⟩ token, and similarly, the target sentence and its immediately preceding sentence are also concatenated with a ⟨CONCAT⟩ token. By translating the concatenated source sentence pair, a pair of translated target sentences concatenated with a ⟨CONCAT⟩ token is obtained. Then, only the second sentence after the ⟨CONCAT⟩ token is extracted and evaluated. In context translation, this 2-to-2 (Tiedemann and

[1]Haddow et al. (2018) evaluated a similar multi-head and multi-hop attention mechanism, although Haddow et al. (2018) employed the vector concatenation over the multiple heads in stead of normalization. Haddow et al. (2018) also reported that the multi-head and multi-hop attention mechanism outperformed the baseline RNN model in the evaluation of the language pairs of CS-EN, EN-CS, ET-EN, EN-ET, FI-EN, and EN-FI, where the length of the training sentences is limited to 50 words or less. In this paper, on the other hand, in the evaluation of the language pairs of JA-EN and EN-JA, the proposed multi-head and multi-hop attention mechanism outperformed the Transformer when the number of tokens is 120-129.

Table 2: Evaluation Result

Model	head	hop	OpenSubtitles 2018		ASPEC		OpenSubtitles 2018 with context	
			ja→en	en→ja	ja→en	en→ja	ja→en	en→ja
RNN baseline	1	1	12.12	9.27	26.41	36.39	13.85	10.24
Multi-head RNN (single-hop attention)	2	1	12.38‡	9.36†	26.63	36.60†	14.14‡	10.32
	3	1	12.42‡	9.55‡	26.98†	36.55	14.28‡	10.53‡
Proposed Method (multi-hop independent attention)	2	2	12.47‡	9.61‡	26.95†	36.31	14.16‡	10.18
Proposed Method (multi-hop dependent attention)	2	2	12.87‡	9.89‡	27.33‡	36.91‡	14.41‡	10.74‡
	2	3	12.88‡	9.87‡	27.39‡	37.41‡	14.79‡	10.79‡
	3	2	13.03‡	9.83‡	27.27‡	37.54‡	14.83‡	10.55‡
	3	3	13.03‡	9.76‡	27.21‡	37.49‡	14.52‡	10.76‡
Transformer	4	1	15.20	10.95	27.50	38.25	15.98	11.44

Proposed methods that significantly outperform the RNN baseline are indicated by †($p \leq 0.05$) and ‡($p \leq 0.01$).



[1]ja→en [2]en→ja

Figure 3: BLEU per sentence length (ASPEC)

Scherrer, 2017) method is a major and increases the number of length per sentence. So, context translation faces long sentence translation.

For ASPEC, among the 3,000,000 training sentence pairs, 1,000,000 sentence pairs with the highest sentence alignment scores were used. Other than the training sentence pairs, 1,790 sentence pairs as the development data as well as 1,812 sentence pairs as the test data are provided by Nakazawa et al. (2016). Also, held out sentence pairs other than those training/development/test data sets are used for the evaluation per sentence length in Section 4.3.

For ASPEC, we conducted an evaluation per sentence length. The widths of the sentence length are segmented with the intervals of 10 words such as 0-9 words, 10-19 words, . . ., etc. Each subset for a range of the sentence length is constructed by collecting sentences within that range according to the criterion that the total number of word tokens within each subset is kept as 20,000. Here, for several subsets of short sentences as well as long sentences, held out development sentence pairs with the highest sentence alignment scores are used so as to keep the total number of word tokens within each subset as 20,000. We do not set any upper bound of sentence length in training/development/test. This is for the purpose of evaluating the capability of the proposed method against long sentences.

For tokenization, we used the SentencePiece tool (Kudo and Richardson, 2018) to set the vocabulary size of 32,000 each for both Japanese and English in order to avoid unknown words. Before splitting into subword units by SentencePiece, tokenization is performed by the morphological analysis tool MeCab[2] for Japanese, and by Moses Tokenizer (Koehn et al., 2007) for English[3].

[2] http://taku910.github.io/mecab/

[3] By performing tokenization before splitting into subword

Table 3: BLEU per sentence length (ASPEC ja→en)

sentence length	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-89	90-99	100-109	110-119	120-129	130-139
number of sentences	1594	1248	810	579	457	372	315	272	238	214	192	176	162	151
RNN baseline	5.94	18.47	27.10	27.16	25.44	22.97	23.71	21.70	21.34	20.96	23.14	21.18	19.78	18.73
multi-hop dependent (head2, hop2)	6.40‡	19.43‡	27.62	27.78	26.49†	24.08‡	25.02‡	22.42	22.74‡	22.72‡	23.11	20.62	20.56††	20.62‡††
Transformer	7.50	19.70	27.29	28.83	25.67	23.62	24.31	22.39	22.34	21.90	22.89	20.80	17.88	17.36

Proposed methods that significantly outperform the RNN Baseline are indicated by †($p \leq 0.05$) and ‡($p \leq 0.01$). Proposed methods that significantly outperform the Transformer are indicated by †($p \leq 0.05$) and ††($p \leq 0.01$).

Table 4: BLEU per sentence length (ASPEC en→ja)

sentence length	0-9	10-19	20-29	30-39	40-49	50-59	60-69	70-79	80-89	90-99	100-109	110-119	120-129	130-139
number of sentences	2531	1303	823	591	458	373	314	272	239	213	193	177	162	108
RNN baseline	24.44	32.41	36.60	36.16	35.58	33.29	31.75	28.65	29.72	28.43	27.14	25.21	20.82	16.86
multi-hop dependent (head2, hop2)	24.90†	32.83	37.03	37.28‡	35.91	33.17	33.40‡	30.12†	30.54	28.52	26.33	25.43	24.18‡††	20.01†
Transformer	25.06	34.41	38.79	38.69	37.09	34.10	33.95	31.49	32.35	29.49	27.00	24.43	21.90	18.22

4.2 Experimental Setup

The baseline is the bidirectional sequence-to-sequence model (Luong et al., 2015) using Long Short-Term Memory (LSTM) which is a kind of RNN. We used fairseq (Gehring et al., 2017) for implementation.

As training, we used Nesterov's Accelerated Gradient (Sutskever et al., 2013) as optimizer with a learning rate of 0.005. The embedding size was 512, the hidden size was 1024, and the encoder and the decoder are of one layer each. For comparison, we also conducted evaluation with the Transformer, where the number of heads was set to 4 according to the default setting[4] of fairseq, and its learning rate was set to 0.0001 following the result of investigating the value at which its loss converged. For all the models, the number of epochs in training was 20. The number of tokens per batch was 2,000 and two GPUs were used in parallel[5].

4.3 Result

Evaluation results are shown in Table 2. Hereafter, as the proposed method without any specific notice, we refer to the model with two heads and two hops of multi-hop dependent attention, which is the model described in Section 3 and Figure 2.

In the evaluation of Japanese-to-English translation of ASPEC, the BLEU of the proposed method was 27.33, which significantly outperforms 26.41 BLEU of RNN baseline. And English-to-Japanese translation of ASPEC, the BLEU of the proposed method was 36.91, which significantly outperforms 36.39 BLEU of RNN baseline. In addition to that, when we measured BLEU for each sentence length, the proposed method significantly outperforms Transformer when the sentence length was between 120 and 129 tokens both direction (Figure 3 [1], Table 3, Figure 3 [2], Table 4). Also, there is no long sentence which has over 120 tokens in the English side of the training corpus.

In multi-hop dependent attention, each head used the information of another head when calculating secondary attention, and two heads shared their parameters. We also evaluated the multi-hop independent attention, where their two heads do not share any information. According to ASPEC's Japanese-to-English translation, the multi-hop dependent attention model achieved the BLEU of 27.33, while the BLEU of the multi-hop independent attention model was 26.95. In the English-to-Japanese translation, the dependent model achieved the BLEU of 36.91, while that of the independent model was 36.31. Both differences are significant at the level of 1% respectively.

In addition, the single-hop attention refers to a model that introduces multi-head attention into source-target attention of RNN and simply increases the number of heads. In the single-hop model with two heads, the BLEU in the evaluation of Japanese-to-English translation of ASPEC was 26.63, which was lower than that of the proposed multi-hop dependent attention model as 27.33. The single-hop attention model is inferior to the proposed multi-hop dependent attention model for all the data sets and both translation directions. Thus, this result supports the usefulness of the proposed multi-hop dependent attention model.

units by SentencePiece, it is guaranteed that any subword unit concatenating over tokenization boundaries is avoided.

[4]Its embedding size is 512, its hidden size is 512, the optimizer used is adam, the encoder and the decoder are of 6 layers each.

[5]The speed of the decoder of the proposed multi-head and multi-hop dependent attention model is roughly two-thirds of that of the baseline RNN model where the numbers of heads and hops are 2.

Table 5: Model Parameters

Model	head	hop	Parameter
RNN baseline	1	1	68,460,544
Multi-head RNN (single-hop attention)	2	1	70,557,696
	3	1	72,654,848
Proposed Method (multi-hop independent attention)	2	2	72,654,848
Proposed Method (multi-hop dependent attention)	2	2	75,800,576
	2	3	81,043,456
	3	2	79,994,880
	3	3	87,334,912
Transformer	4	1	81,604,608

5 Related Works

Dehghani et al. (2019) proposed Universal Transformer for solving the problems of Transformer including the weakness for long distance dependency. Although it has a mechanism to repeat updating the states for each word with parameters shared, it requires a larger number of parameters than Transformer. There could be an approach like BERT (Devlin et al., 2019) where the number of parameters is increased significantly to make a more powerful Transformer model. Our approach, on the other hand, improves the strength of RNN with a little increase of parameters as shown in Table 5. Moreover, Iida et al. (2019) also applied the multi-hop attention mechanism to the Transformer and reported that the Transformer augmented with the multi-hop attention mechanism significantly outperformed the Transformer. Among other existing approaches to neural machine translation, it is known that ConvS2S (Gehring et al., 2017) is equipped with multiple decoder layers where each decoder layer has a separate attention module. The attention of each of those multiple layers is computed and is then fed to another layer, which then takes the fed information into account when computing its own attention etc. The way those multiple attentions are computed is similar to the multi-head and multi-hop attention mechanism proposed in this paper.

6 Conclusion

We proposed a novel multi-hop and multi-head attention mechanism for RNN encoder-decoder in which each head depends on each other repeatedly. We found that the proposed method significantly outperforms the baseline attention-based RNN encoder-decoder. We also found that it outperforms Transformer when the input sentence is very long.

As we showed in Table 2, among the numbers of multi-head and multi-hop, the pair of the numbers of multi-head and multi-hop with the highest BLEU score varies according to the data sets. Considering this fact, one future work is to study how to estimate the pair of the numbers of multi-head and multi-hop with the optimal BLEU score by introducing a held-out development data set.

References

Bahdanau, D., K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. 3rd ICLR*, pages 42–51.

Bawden, R., R. Sennrich, A. Birch, and B. Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proc. NAACL-HLT*, pages 1304–1313.

Chen, M. X., O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, M. Schuster, N. Shazeer, N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, Z. Chen, Y. Wu, and M. Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proc. 56th ACL*, pages 76–86.

Dehghani, M., S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser. 2019. Universal transformers. In *Proc. 7th ICLR*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*, pages 4171–4186.

Gehring, J., M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proc. 34th ICML*, pages 1243–1252.

Haddow, B., N. Bogoychev, D. Emelin, U. Germann, R. Grundkiewicz, K. Heafield, A. V. M. Barone, and R. Sennrich. 2018. The university of Edinburgh's submissions to the WMT18 news translation task. In *Proc. 3rd WMT, Volume 2: Shared Task Papers*, pages 403–413.

Iida, S., R. Kimura, H. Cui, P. Hung, T. Utsuro, and M. Nagata. 2019. Attention over heads: A multi-hop attention for neural machine translation. In *Proc. 57th ACL, Student Research Workshop*.

Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. 45th ACL, Companion Volume*, pages 177–180.

Kudo, T. and J. Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Libovický, J. and J. Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proc. 55th ACL*, pages 196–202.

Lison, P., J. Tiedemann, and M. Kouylekov. 2018. Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proc. 11th LREC*, pages 1742–1748.

Luong, T., H. Pham, and C. D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*, pages 1412–1421.

Nakazawa, T., M. Yaguchi, K. Uchimoto, M. Utiyama, E. Sumita, S. Kurohashi, and H. Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In *Proc. 10th LREC*, pages 2204–2208.

Papineni, K., S. Roukos, T. Ward, and W. J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. 40th ACL*, pages 311–318.

Sukhbaatar, S., A. Szlam, J. Weston, and R. Fergus. 2015. End-to-end memory networks. In *Proc. 28th NIPS*, pages 2440–2448.

Sutskever, I., J. Martens, George G. Dahl, and G. Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proc. 30th ICML*, pages 1139–1147.

Sutskever, I., O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural machine translation. In *Proc. 27th NIPS*, pages 3104–3112.

Tiedemann, J. and Y. Scherrer. 2017. Neural machine translation with extended context. In *Proc. 2017 DiscoMT*, pages 82–92.

Tran, K., A. Bisazza, and C. Monz. 2018. The importance of being recurrent for modeling hierarchical structure. In *Proc. EMNLP*, pages 4731–4736.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Proc. 30th NIPS*, pages 5998–6008.