

---

# A Target Attention Model for Neural Machine Translation

Hideya Mino † ‡

Andrew Finch †

Eiichiro Sumita †

mino.h-gq@nhk.or.jp

andyfinch@hotmail.com

eiichiro.sumita@nict.go.jp

† National Institute of Information and Communications Technology, 3-5 Hikaridai,  
Seika-cho, Soraku-gun, Kyoto 619-0289, JAPAN

‡ Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Meguro-ku, 13, 152-8550,  
JAPAN

---

## Abstract

Neural Machine Translation (NMT) with an attention mechanism has shown promising results by utilizing word alignments between the source and target sentences. Typically, training of NMT proceeds token-by-token on the target side, where each token is predicted using only a vector representing the current hidden-state, and the previous token. However, this strategy has serious shortcomings originating the lack of information about the partial target sequence hypothesis; specifically, this can lead to source tokens being translated multiple times or remaining untranslated. To alleviate this problem, we introduce a target-side attention mechanism to exploit the generated target sequence of tokens more effectively. We calculate a target-side context vector using a recurrent neural network and feed it to an attention mechanism so that the decoder can pay more or less attention to each token in the partially generated target sequence when predicting the next target token. Experiments on three different English-to-Japanese translation tasks show improvements of 0.6-1.5 BLEU points.

## 1 Introduction

Recently, Neural Machine Translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014) has been growing in popularity due to its capacity to model the translation process end-to-end within a single probabilistic model, and its potential for higher performance compared to existing phrase-based statistical machine translation (SMT) (Koehn, 2004). There are some unique features of NMT models which pose significant challenges for machine translation. One is that NMT systems exploit Long Short-Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997) (or the similar Gated Recurrent Units (GRUs) (Cho et al., 2014)) which allow the systems to capture long-distance dependencies better than vanilla RNNs. Another is the attention mechanism, whereby the decoder can attend directly to localized information from the source sequence of tokens for generating the target sequence (Bahdanau et al., 2015; Luong et al., 2015). NMT systems are generally trained to maximize the likelihood of generating the target sequence of tokens given the source sequence. In practice, each target token is generated conditioned on the vector representing the current hidden-state of the model, and the previously generated target token.

NMT, however, has a serious drawback in that some input tokens are unnecessarily translated or mistakenly left untranslated (Tu et al., 2016). Our hypothesis is that this is mainly

because the hidden state of the LSTM decoder is not sufficiently representing all the information concerning the generated target sequence of tokens. Our work therefore endeavors to alleviate this drawback by explicitly handing a summary of the target sequence generated so far, at each step in the decoding process. Although an LSTM is able to provide the function of a long-term memory, the prediction of target tokens in a state-of-the-art NMT model (Bahdanau et al., 2015) heavily depends on two factors: the source-side context vectors with focus provided by an attention model, and a target language model implicitly learned by the LSTM decoder. This NMT model fails to exploit the generated target-side information, which is useful to avoid over- and under-translation problems. If target words translated in the past is accumulated appropriately to the LSTM decoder, they are less likely to be translated again, and new target word which is not translated yet should be generated. Because of ignoring the information of the sequence of previously generated target tokens, unnecessarily translated words and mistakenly untranslated words are generated. To alleviate the lack of target-side information in the LSTM decoder, we propose to add a target-side context vector directly into the NMT model. The target-side context vector is generated with the attention mechanism, which selects the relevant target tokens for predicting the next target token. We show empirically that the addition of this target-side context vector significantly improves the performance of an NMT system on three different English-to-Japanese translation tasks.

## 2 Related Work

There is much recent work on augmenting attention-based NMT systems with additional features. One focus is the use of the monolingual data (Sennrich et al., 2016; Gülçehre et al., 2015). Gülçehre et al. (2015) incorporated a large language model into an attention-based NMT system to allow the effective use of target-side monolingual data. Another focus is in designing better decoding strategies (Luong et al., 2015; Tu et al., 2016; Mi et al., 2016; Liu et al., 2016; Mi et al., 2016; Tu et al., 2017). Tu et al. (2017) proposed to augment a direct model’s decoding objective with a reverse translation model. Liu et al. (2016) proposed translating in both a left-to-right and a right-to-left direction and seeking a consensus. Tu et al. (2016) introduced a coverage vector to keep track of the attention history, which encourages the attention-based NMT system not to translate source words for multiple times (i.e., avoiding over-translation) and to translate more untranslated source words (i.e., avoiding under-translation). Mi et al. (2016) also dealt with the coverage problem.

We also tackle on the over- and under-translation problems. Our approach differs from those of Tu et al. (2016) and Mi et al. (2016) in that they utilize only source-side attention history, whereas our approach also exploits the sequence of target tokens generated.

## 3 Neural Machine Translation with a Source Attention Model

Our method is based on NMT with attention (Bahdanau et al., 2015), which generates the target sentence  $\mathbf{y} = (y_1, \dots, y_M)$  from the source sentence  $\mathbf{x} = (x_1, \dots, x_N)$  of length  $N$ , as illustrated in Figure 1 (note: we use bold script to denote sequences hereafter). The attention-based model consists of two components, an encoder and a decoder. The encoder reads the source sentence  $\mathbf{x}$  and encodes it into hidden states  $\mathbf{h} = (h_1, \dots, h_N)$ . The hidden states are produced using a bidirectional RNN, which concatenates a forward and a backward sequences, as

$$h_j = \left[ \begin{array}{c} \vec{h}_j \\ \overleftarrow{h}_j \end{array} \right] \quad (1)$$

where

$$\vec{h}_j = e_1(x_j, \vec{h}_{j-1}), \overleftarrow{h}_j = e_2(x_j, \overleftarrow{h}_{j+1}). \quad (2)$$

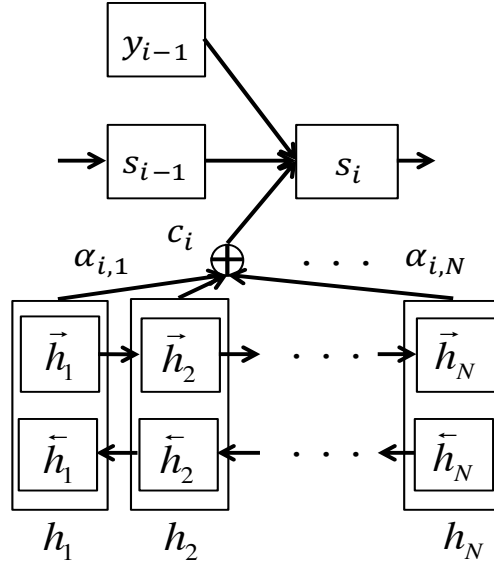


Figure 1: Encoder-decoder NMT architecture with source attention

$e_1$  and  $e_2$  are nonlinear functions. Bahdanau et al. (2015) used a GRU (Cho et al., 2014) for  $e_1$  and  $e_2$ . Each hidden state, represented as a single vector, includes not only the lexical information at its source position, but also information about the unbounded length of the left and right context. Then, the decoder predicts the target sentence  $\mathbf{y}$  using a conditional probability calculated as

$$p(y_i | \mathbf{y}_{1:i-1}, \mathbf{x}) = f_1(y_{i-1}, s_i, c_i) \quad (3)$$

where  $\mathbf{y}_{1:i-1}$  is a partial translation  $(y_1, \dots, y_{i-1})$ ,  $f_1$  is implemented as a feedforward neural network with a softmax output layer,  $s_i$  is a hidden state of the RNN, and  $c_i$  is a context vector derived from the source sentence. The hidden state  $s_i$  of the target RNN is computed by

$$s_i = g_1(s_{i-1}, y_{i-1}, c_i) \quad (4)$$

where  $g_1$  is a nonlinear function analogous to  $e_1$  or  $e_2$ . The context vector  $c_i$  is computed as a convex sum of the hidden states  $h_j$  of Equation (1):

$$c_i = \sum_{j=1}^N \alpha_{i,j} h_j \quad (5)$$

where  $\alpha_{i,j}$  is a scalar weight of each hidden state  $h_j$  computed by

$$\alpha_{i,j} = \frac{\exp\{a(s_{i-1}, h_j)\}}{\sum_{k=1}^N \exp\{a(s_{i-1}, h_k)\}} \quad (6)$$

where  $a$  is a feedforward neural network with a single hidden layer. The attention mechanism is driven by this  $\alpha_{i,j}$ , which shows how well the input context at the  $j$ -th word and the output word at the  $i$ -th position match. The objective is to jointly maximize the conditional probability

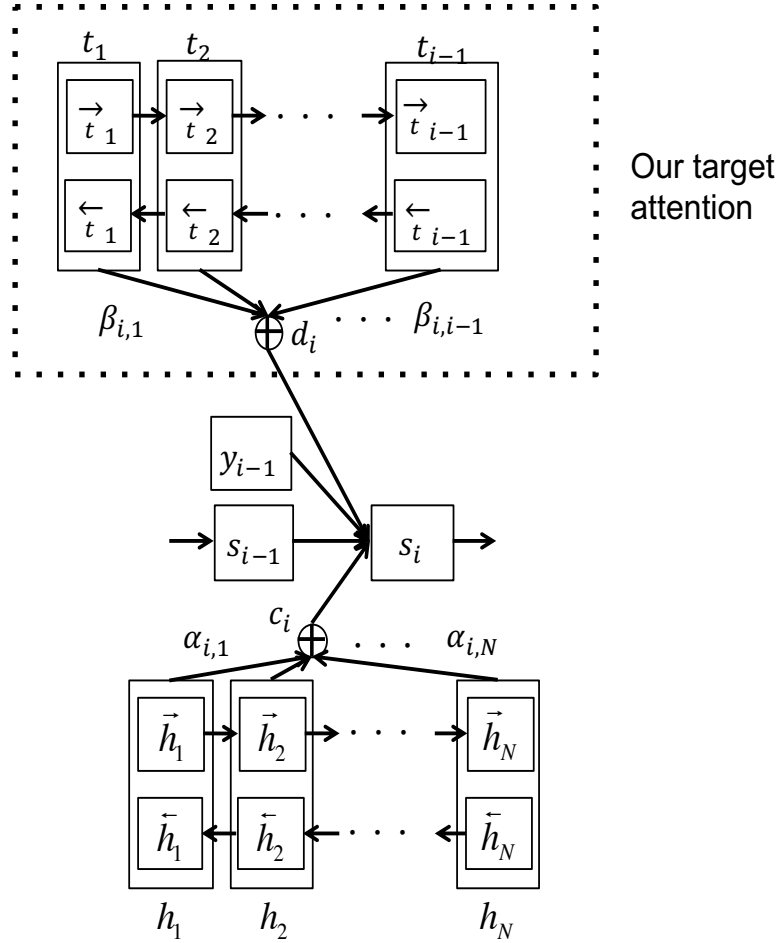


Figure 2: The proposed encoder-decoder NMT architecture with both source and target attention

for each generated target word as

$$\theta^* = \arg \max_{\theta} \sum_{k=1}^K \sum_{i=1}^{M_k} \log p(y_i^k | \mathbf{y}_{1,i-1}^k, \mathbf{x}^k, \theta) \quad (7)$$

where  $(\mathbf{x}^k, \mathbf{y}^k)$  is the  $k$ -th training pair of sentences, and  $M_k$  is the length of the  $k$ -th target sentence  $\mathbf{y}^k$ .

#### 4 Adding a Target Attention Model

Attention-based NMT usually uses an LSTM for decoding from an encoded source sentence as a whole, and a single previous target token as in Equation (4). Intuitively, the encoded source sentence and the generated sequence of target tokens are both indispensable for predicting the next target token. Although LSTMs have been shown to be capable of predicting the next token in a sequence given a compressed representation of the preceding sequence, this process becomes considerably more difficult when compressing long sequences (Liu et al., 2016). To

strengthen the information provided by the generated target sequence of tokens, our model adds a target-side context vector to the input of the LSTM decoder at each decoding step, as shown in Figure 2. In this model, a representation of the generated target sequence is explicitly made available to the decoder at each step instead of implicitly relying on the LSTM to maintain it.

In addition, the semantics each token of the generated target sequence depends on its context. The LSTM model produces a vector that contains compressed information representing an unfocused summary of the whole generated target sequence. In order to allow the model to focus on salient contexts, we use a mechanism for focusing on the relevant parts of the already-generated target sequence for generating the current target token, along with a bidirectional layer to provide the model with the a good representation of the target.

The proposed method is implemented as a target-side attention model constructed analogously to the source-side attention model, where the attention ranges over the partially generated target token sequence. More formally, the partial translation  $\mathbf{y}_{1,i-1}$  is encoded into a sequence of hidden states  $\mathbf{t}_{1,i-1}$ , which are produced using a bidirectional RNN, as

$$t_k = \begin{bmatrix} \overrightarrow{t}_k \\ \overleftarrow{t}_k \end{bmatrix} \quad (1 \leq k \leq i-1) \quad (8)$$

where

$$\overrightarrow{t}_i = e_3(y_i, \overrightarrow{t}_{i-1}), \overleftarrow{t}_i = e_4(y_i, \overleftarrow{t}_{i+1}). \quad (9)$$

$e_3$  and  $e_4$  are nonlinear functions as in Equation (2). Then, the decoder predicts the target sentence with a conditional probability as

$$p(y_i | \mathbf{y}_{1,i-1}, \mathbf{x}) = f_2(y_{i-1}, s_i, c_i, d_i) \quad (10)$$

where  $f_2$  is a probability estimator as in Equation (3) and newly introduced  $d_i$  is a predicted target-side context vector. The computation of the hidden state  $s_i$  is also modified as

$$s_i = g_2(s_{i-1}, y_{i-1}, c_i, d_i) \quad (11)$$

where  $g_2$  is a nonlinear function as in Equation (4). The context vector  $d_i$  is computed as a convex sum of the hidden states  $\mathbf{t}_{1,i-1}$ :

$$d_i = \sum_{k=1}^{i-1} \beta_{i,k} t_k \quad (12)$$

where  $\beta_{i,k}$  is also a scalar weight of each hidden state  $t_k$  as below:

$$\beta_{i,k} = \frac{\exp\{b(s_{i-1}, t_k)\}}{\sum_{k=1}^{i-1} \exp\{b(s_{i-1}, t_k)\}} \quad (13)$$

where  $b$  is a feedforward neural network analogous to  $a$  in Equation (6).  $\beta_{i,k}$  gives a normalized score for each previous target token, which measures how the  $k$ -th target word is relevant to the prediction of the  $i$ -th target token. The objective is again to jointly maximize the likelihood as in Equation (7). Typically, the previous target token  $y_{i-1}$  used by the LSTM decoder is the true previous token when training, and a predicted previous token during decoding. In our experiments, we follow this practice, although there is evidence that using predictions during training would be beneficial (Bengio et al., 2015). Since our approach is orthogonal to that of Bengio et al. (2015), it would be possible to use both techniques in tandem.

Corpus	Training					Development			Test		
	Sents.	Word types		Avg. length		Sents.	Word types		Sents.	Word types	
		en	ja	en	ja		en	ja		en	ja
IWSLT'07	40k	9.4k	10k	9.3	12.7	0.5k	1.2k	1.3k	0.5k	0.8k	0.9k
NTCIR-10	717k	105k	79k	23.3	27.7	2.0k	5.0k	4.4k	0.5k	2.4k	2.1k
ASPEC	843k	288k	143k	22.1	23.9	1.8k	7.1k	6.3k	1.8k	7.0k	6.4k

Table 1: Data sets

## 5 Experiments

We evaluated the proposed method on three different English-to-Japanese translation tasks. As a baseline, we trained the attention-based NMT and the coverage-vector method (Tu et al., 2016). To confirm the effectiveness of the target-side bidirectional RNN in the proposed method, we also trained the proposed method with one direction RNN, from left to right.

### 5.1 Data and model parameters

The corpora we used were IWSLT'07 (Fordyce, 2007), NTCIR-10 (Goto et al., 2013), and ASPEC (Nakazawa et al., 2016). IWSLT'07 consists of spoken travel conversations, NTCIR-10 consists of patents, and ASPEC is in the domain of scientific publications. We constrained training sentences to have a maximum length of 40 to speed up the training.<sup>1</sup> As shown in Table 1, the data size of IWSLT'07 is smaller than the other corpora, and ASPEC has a greater lexical variety compared to the others. Each test sentence had a single reference translation. The English data was tokenized using the tokenization script included in the Moses decoder.<sup>2</sup> The Japanese data was tokenized with KyTea (Neubig et al., 2011).

### 5.2 Settings

The input and output of our model are sequences of one-hot vectors with dimensionality corresponding to the sizes of the source and target vocabularies. For NTCIR-10 and ASPEC, we replaced words of frequency less than 3 with the *[UNK]* symbol and excluded them from the vocabularies. As a result, the number of word types in NTCIR-10 turned out 60k for English and 50k for Japanese, and ASPEC contained 124k types for English and 79k for Japanese. Due to the limited memory of GPU, each source and target word was projected into a 200-dimensional continuous Euclidean space to reduce the dimensionality, the depth of the stacking LSTMs was 1 and hidden layer size was set to 300. Each model was optimized using Adam (Kingma and Ba, 2014) with the following parameters:  $\alpha = 1e - 3$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 1e - 8$ . To prevent overfitting we used dropout (Srivastava et al., 2014) with a drop rate of  $r = 0.5$  to the last layer of each stacking LSTM. All weight metrics of each model were initialized by sampling from a normal distribution of zero mean and 0.05 standard deviation. The gradient at each update is calculated using a minibatch of at most 100 sentence pairs and we ran for a maximum of 30 iterations for the entire training data. Training was early-stopped to maximize the performance on the development set measured by BLEU. We used a single Tesla K80 GPU with 12 GB of memory for the training. For decoding, we used beam search with a beam size of 10. The beam search was terminated when an end-of-sentence *[EOS]* symbol was generated.

The evaluation metric is case-insensitive BLEU (Papineni et al., 2002) calculated by the

<sup>1</sup>The proposed method takes approximately five times the training time, and three times the decoding time, relative to the baseline attention-based NMT. The proposed method with one direction RNN, instead of bidirectional RNN, takes approximately three times the training time, and three times the decoding time.

<sup>2</sup><http://statmt.org/moses/>

System	IWSLT'07	NTCIR-10	ASPEC
<i>source-attn</i>	47.4	31.0	26.2
<i>coverage-vector</i>	47.7	31.4	25.8
<i>source-and-target-attn (left-to-right)</i>	48.0	31.5	26.4
<i>source-and-target-attn (bidirectional)</i>	48.3	32.3 †‡	27.7 †‡

Table 2: BLEU scores for the attention-based NMT (*source-attn*), the coverage vector method (Tu et al., 2016) (*coverage-vector*) and the proposed method (*source-and-target-attn*) with target-side bidirectional RNN (*bidirectional*) and target-side one directional RNN from left to right (*left-to-right*) (†: significantly better than *source-attn* ( $p < 0.05$ ); ‡: significantly better than *coverage-vector* ( $p < 0.05$ )).

System	IWSLT'07	NTCIR-10	ASPEC
<i>source-attn</i>	39 / 0.91	412 / 0.94	1178 / 0.91
<i>coverage-vector</i>	91 / 0.91	347 / 0.92	884 / 0.89
<i>source-and-target-attn (left-to-right)</i>	58 / 0.91	286 / 0.93	870 / 0.90
<i>source-and-target-attn (bidirectional)</i>	38 / 0.90	335 / 0.94	659 / 0.91

Table 3: Numbers of overtranslated words (left-side) and averages of the brevity penalty per sentence (right-side)

`multi-bleu.perl` script in the Moses toolkit. Statistical significance testing of the BLEU differences was performed using paired bootstrap resampling (Koehn, 2004) with 10,000 iterations. We also assessed the decrease in the over- and under-translation with two kinds of criteria. For the over-translation, we used a number of overtranslated words, which are unnecessarily translated though these are already translated in outputs. We simply counted the number of repeated phrases (length longer or equal than 2 words) for each sentence as in Mi et al. (2016). For the under-translation, we used an average of brevity penalty per sentence. The brevity penalty, which is part of BLEU, is to penalize predicted sentence that are shorter than the reference.

### 5.3 Results

Table 2 summarizes the results for all the three tasks. For the IWSLT'07 task, our model achieved 0.9, 0.6, and 0.3 BLEU point improvements compared with *source-attn*, *coverage-vector*, and *source-and-target-attn (left-to-right)*, respectively. For the NTCIR-10 task, our model achieved gains of 1.3, 0.9, and 0.8 BLEU points. For the ASPEC task, our model achieved gains of 1.5, 1.9, and 1.3 BLEU points. These results show that our proposed method is more effective than other baseline methods. The results for IWSLT'07 show less improvement than those for NTCIR-10 and ASPEC. The reason for this may be the length of the target. As shown in Table 1, the average length of sentence of IWSLT'07 is much shorter than NTCIR-10 and ASPEC. These results show that the proposed method seems to be more effective for the tasks with long sentences. The explanation is most likely analogous to the motivation for using a source-side attention model: an LSTM model without attention struggles to propagate necessary information over longer distances. Our target-side attention model explicitly facilitates this.

Table 3 shows the numbers of overtranslated words and the averages of the brevity penalty. The brevity penalty is 1.0 when the output length is longer than the reference translation's length. For IWSLT'07, there were no improvements. As mentioned earlier, we believe the cause is related to the fact that the sentences in this corpus are short; our method is most ef-

fective for longer sequences. For the other two tasks, our model seemed to be able to reduce the number of overtranslated words, also maintaining the target sequence length closer to that of the references. For NTCIR-10, though *source-and-target-attn (left-to-right)* greatly reduces the number of overtranslated words, the BLEU score is almost same as *coverage-vector*. It shows that *source-and-target-attn (left-to-right)* increases the number of mistranslated words and *source-and-target-attn (bidirectional)* is effective to decrease not only the number of overtranslated words but also the number of mistranslated words. Examples of outputs generated by each model are shown in Appendix A.

These analyses validate our contribution to the original motivation for this work, i.e., the proposed model is capable of effectively decreasing the number of mistakenly untranslated words and unnecessarily translations of the same word.

## 6 Conclusion

We introduced a focused summary of the target sequence generated so far into the decoding process in order to alleviate the problems of the over- and under-translation problems. Our empirical evaluation shows that the proposed method is effective in achieving substantial improvements in terms of translation quality consistently across three different tasks.

## Acknowledgements

We are deeply grateful to Atsushi Fujita and anonymous reviewers for their suggestions and insightful comments on the early version of this paper.

## References

- Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR 2015)*.
- Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. M. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems (NIPS 2015)*.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*.
- Fordyce, C. S. (2007). Overview of the 4th international workshop on spoken language translation iwslt 2007 evaluation campaign. In *Proceedings of the 4th International Workshop on Spoken Language Translation (IWSLT 2007)*.
- Goto, I., Chow, K., Lu, B., Sumita, E., and Tsou, B. K. (2013). Overview of the patent machine translation task at the NTCIR-10 workshop. In *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR-10)*.
- Gülçehre, Ç., Firat, O., Xu, K., Cho, K., Barrault, L., Lin, H., Bougares, F., Schwenk, H., and Bengio, Y. (2015). On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8).
- Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.



- Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*.
- Liu, L., Utiyama, M., Finch, A., and Sumita, E. (2016). Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016)*.
- Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*.
- Mi, H., Sankaran, B., Wang, Z., and Ittycheriah, A. (2016). Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*.
- Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S., and Isahara, H. (2016). Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*.
- Neubig, G., Nakata, Y., and Mori, S. (2011). Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL 2011)*.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*.
- Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15.
- Sutskever, I., Vinyals, O., and Le, Q. V. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS 2014)*.
- Tu, Z., Liu, Y., Shang, L., Liu, X., and Li, H. (2017). Neural machine translation with reconstruction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*.
- Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016). Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.

## **Appendix A. Examples of outputs**

We show examples of Japanese translation generated with each of the four models in Tables 2 and 3 with a source sentence and a reference. The words shown in bold letters are examples of over- or under-translation problems.

## Examples of NTCIR-10

### [Example 1]

Source sentence:

This fluctuation in the power supply voltage and reference voltage causes power source noise .

Reference:

このようにして電源電圧 (**the power supply voltage**) や基準電圧 (**the reference voltage**) が変動して電源ノイズを生じさせる。

Output with *source-attn*:

電源電圧 (**the power supply voltage**) と電源電圧 (**the power supply voltage**) との変動により、電源ノイズが発生する。

Output with *coverage-vector*:

電源電圧 (**the power supply voltage**) の変動により、電源電圧 (**the power supply voltage**) が変動し、電源電圧 (**the power supply voltage**) が発生する。

Output with *source-and-target-attn (left-to-right)*:

電源電圧 (**the power supply voltage**) および基準電圧 (**the reference voltage**) の変動は、電源ノイズを発生する。

Output with *source-and-target-attn (bidirectional)*:

電源電圧 (**the power supply voltage**) と基準電圧 (**the reference voltage**) との変動は、電源ノイズを発生する。

### [Example 2]

Source sentence:

As shown in FIG . 5 , the drain current is also affected by the stress .

Reference:

図5に示したようにドレイン電流 (**the drain current**) も応力の影響を受ける。

Output with *source-attn*:

5 . 5 に示すように、ドレイン電流 (**the drain current**) の影響を受けることにより、ドレイン電流 (**the drain current**) が影響を受ける。

Output with *coverage-vector*:

5 v に示すように、ドレイン電流 (**the drain current**) によりドレイン電流 (**the drain current**) も影響を受ける。

Output with *source-and-target-attn (left-to-right)*:

図5に示すように、ドレイン電流 (**the drain current**) は、応力によって影響を与える (**is affecting**) 。

Output with *source-and-target-attn (bidirectional)*:

図5に示すように、ドレイン電流 (**the drain current**) は、応力によって影響を受ける (**is affected by**) 。

## Examples of ASPEC

### [Example 1]

Source sentence:

compatible solutes include polyols such as glycine betaine ( betaine ), zwitterionic proline , pinitol , sorbitol , and mannitol .

Reference:

適合 溶質 としては、グリシンベタイン（ベタイン）、プロリンなどの両性イオン (zwitterionic)、ピニトール (pinitol)、ソルビトール (sorbitol)、マンニトールなどのポリオール類などが知られている。

Output with *source-attn*:

適合 溶質 はグリシンベタイン（グリシンベタイン）、ピニトール (pinitol)、ピニトール (pinitol)、ピニトール (pinitol)、ピニトール (pinitol) である。

Output with *coverage-vector*:

適合 溶質 としては、グリシンベタイン（ベタイン）、ピニトール (pinitol)、ピニトール (pinitol)、ピニトール (pinitol)、ピニトール (pinitol) である。

Output with *source-and-target-attn (left-to-right)*:

適合 溶質 としてはグリシンベタイン（ベタイン）、両性イオン (zwitterionic)、ピニトール (pinitol)、ピニトール (pinitol)、マンニトールなどのポリオール類が含まれる。

Output with *source-and-target-attn (bidirectional)*:

適合 溶質 としては、グリシンベタイン（ベタイン）、ピニトール (pinitol)、ソルビトール (sorbitol)、マンニトールなどのポリオールを含む。

### [Example 2]

Source sentence:

the liquid-crystal film in the title was prepared on a glass substrate by uniaxial orientation and the amount of the transmitted light through a polarizing plate was measured .

Reference:

標題 液晶 薄膜 をガラス 基板 上に (on a glass substrate) 一軸 配向 させて 形成 し、偏光板 を 経た 透過 光 量を 測定 した。

Output with *source-attn*:

標記 薄膜 をガラス 基板 上に (on a glass substrate) ガラス 基板 上に (on a glass substrate) 調製 し、偏光板 を 通過 する 透過 光 の 量 を 測定 した。

Output with *coverage-vector*:

標記 薄膜 をガラス 基板 上に (on a glass substrate) ガラス 基板 上に (on a glass substrate) 作製 し、偏光板 を 通過 する 透過 光 の 量 を 測定 した。

Output with *source-and-target-attn (left-to-right)*:

標記液晶膜を一軸配向によりガラス基板上に (**on a glass substrate**) 作製し、偏光板を介した透過光の量を測定した。

Output with *source-and-target-attn (bidirectional)*:

標記液晶をガラス基板に (**on a glass substrate**) 一軸配向により作製し、偏光顕微鏡により透過光の量を測定した。