Context-aware Neural Machine Translation with Coreference Information

Takumi Ohtani[†], **Hidetaka Kamigaito**[†],
Masaaki Nagata[‡] and **Manabu Okumura**[†]
[†] Institute of Innovative Research, Tokyo Institute of Technology
[‡] NTT Communication Science Laboratories, NTT Corporation
{ohtani, kamigaito}@lr.pi.titech.ac.jp, masaaki.nagata.et@hco.ntt.co.jp, oku@pi.titech.ac.jp

Abstract

We present neural machine translation models for translating a sentence in a text by using a graph-based encoder which can consider coreference relations provided within the text explicitly. The graph-based encoder can dynamically encode the source text without attending to all tokens in the text. In experiments, our proposed models provide statistically significant improvement to the previous approach of at most 0.9 points in the BLEU score on the OpenSubtitle2018 English-to-Japanese data set. Experimental results also show that the graph-based encoder can handle a longer text well, compared with the previous approach.

1 Introduction

The quality of machine translators has recently dramatically improved with Sequence-to-Sequence (Seq2Seq) models (Bahdanau et al., 2014). Most Seq2Seq models are used based on the premise that each sentence is independently translated one by one. In contrast to this premise, real sentences are often an element of a larger unit, such as a document. This means that a sentence is not always semantically self-contained in itself. To correctly interpret a sentence which is a part of a document, it is important to consider its context, preceding and/or succeeding sentences.

In order to tackle the problem, Seq2Seq models that can receive two sentences (Tiedemann and Scherrer, 2017; Bawden et al., 2018; Voita et al., 2018; Wang et al., 2017) have been utilized. For capturing multiple-sentence information more effectively, Miculicich et al. (2018); Zhang et al. (2018) incorporated document-level attention modules into Seq2Seq models. Stojanovski and Fraser (2018) proposed a Seq2Seq model which can capture antecedents of pronouns in the previous source sentence by using a coreference resolution toolkit. To capture the entire source text information, these models strongly depend on attention distributions.

However, the space complexity of the attention mechanism in the Seq2Seq model increases in proportion to the square of the input sequence length, because it tries to attend to all the words in the source text. This characteristic prevents the model from translating a long text. Furthermore, in translating into a pro-drop language such as Japanese, longer contexts are required to generate accurate and naturally concise sentences.

To avoid the problem, we propose a model that can effectively capture contextual information, preceding and succeeding sentences of the source sentence to be translated, by constructing an encoder that is based on explicit coreference relations. The proposed model can directly take into account relationships between sentences via a graph structured encoder constructed with a coreference resolution toolkit. Therefore, it does not need to attend to all input tokens. This characteristic enables our proposed model to handle more sentences in a step, compared with the previous models, and it may improve translation quality when a source text has many sentences.

Experimental results on English-to-Japanese translation pairs in OpenSubtitles2018 (Lison et al., 2018) show that our proposed model can significantly improve the previous model in terms of BLEU scores. In addition, we observe that our model is especially effective in translating a sentence which is a part of a long text, compared to the previous model.

2 Sequence-to-Sequence Model

In this section, we explain the standard Seq2Seq model proposed by Bahdanau et al. (2014), which

Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019), pages 45–50
Hong Kong, China, November 3, 2019. ©2019 Association for Computational Linguistics

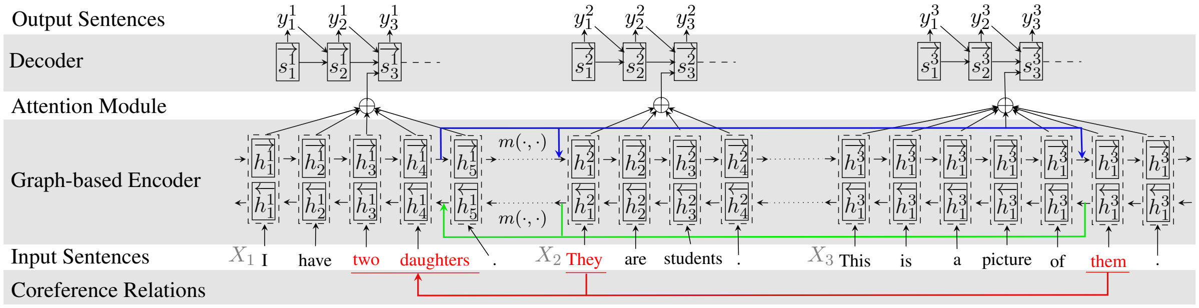

Figure 1: Network structure of the proposed model. Blue arrows indicate a forward hidden state merge operation and green arrows indicate a backward hidden state merge operation. Both operations are based on a coreference relation represented as red arrows. Attention distributions are calculated only on a currently translating sentence.

our proposed model is based on. We use LSTM (Hochreiter and Schmidhuber, 1997) as recurrent neural network (RNN) structures in the encoder and the decoder. In the Seq2Seq model, a probability of translating an input sentence $\mathbf{x} = (x_1, \cdots, x_{T_x})$ into an output sentence $\mathbf{y} = (y_1, \cdots, y_n)$ is represented as follows:

$$\begin{aligned} p\left(y_i | y_1, \ldots, y_{i-1}, \mathbf{x}\right) &= softmax(g\left(s_i, d_i\right)), \\ s_i &= dec\left(s_{i-1}, emb(y_{i-1}), d_i\right), \\ d_i &= \sum_{j=1}^{T_x} a\left(s_{i-1}, h_j\right) h_j, \\ h_t &= enc\left(emb(x_t), h_{t-1}, h_{t+1}\right), \end{aligned} \tag{1}$$

where i is the position of an output token, t is the position of an input token, $emb(\cdot)$ is a function that returns the embedding of an input word, g is a 2-layer feedforward neural network (FFNN), dec is a decoder forward-LSTM, enc is an encoder bidirectional-LSTM (Bi-LSTM), and a is a dot attention (Luong et al., 2015) for calculating the attention weight.

3 Graph-based Encoder with Coreference Relations

Our proposed model can encode not only the sentence to be translated but also its preceding and succeeding sentences together, based on the results of coreference resolution. Therefore, information about sentence relationships can be effectively utilized. Figure1 shows the network structure of our proposed model. At first, input sentences are analyzed by using a coreference resolution system. After that, the encoder part is structured based on the coreference resolution results, and the input text is encoded into hidden states. Then, the hidden states are converted to a translated text via attention distributions and the decoder. During the translation, the attention distributions are only calculated for the currently translated sentence. In the next subsections, we explain the details of each step. We denote a sequence of N sentences as $(X_1, \cdots, X_N)$, and j-th word in X_i as x^i_j hereafter.

3.1 Coreference Resolution

Multiple sentences in a source text $(X_1, \cdots, X_N)$ are concatenated and then input to a coreference resolution system. We use NeuralCoref[1] as the coreference resolution system. Let the length of X_i be T_i. The concatenated token sequence is represented as:

$$(x^1_1, \cdots, x^1_{T_1}, x^2_1, \cdots, x^2_{T_2}, \cdots, x^N_1, \cdots, x^N_{T_N}). \tag{2}$$

The coreference resolution system extracts N_c clusters of coreferring mentions $(c_1, \cdots, c_{N_c})$, which are defined as:

$$c_k = (main_k, sub_k), \tag{3}$$

where $main_k$ is a span of the representative mention in a cluster of coreferring mentions, and sub_k is a span of another mention in the cluster.[2] In general, because many mentions are in a single cluster, the same $main_k$ is sometimes paired to different mentions.

To use coreference relations in our graph-based encoder, we need to consider word-based coreference relations. Let $head(\cdot)$ be a function that returns the first word of an input span and $tail(\cdot)$ be a function that returns the last word of the input span. When x^i_j refers to $x^{i'}_{j'}$, x^i_j and $x^{i'}_{j'}$ satisfy the following conditions:

$$\begin{aligned} x^{i'}_{j'} &= tail(main_k), \\ x^i_j &= head(sub_k). \end{aligned} \tag{4}$$

[1] https://github.com/huggingface/neuralcoref. This code is based on the work by Clark and Manning (2016).

[2] We treat a nominal noun which is the antecedent of a pronoun or a proper noun as a representative mention.

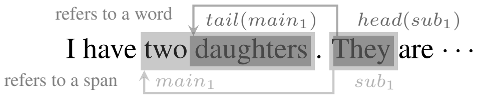

Figure 2: An example of a word-based coreference relation.

Figure 2 shows an example of a word-based coreference relation.

Furthermore, we denote a set of words which are referred by word x^i_j as $ref(x^i_j)$. Because the number of words referred by a word is at most one, the number of elements in $ref(x^i_j)$ is either 1 or 0. $ref(x^i_j)$ can be divided into either anaphora, $ref_f(x^i_j)$, or cataphora, $ref_b(x^i_j)$, as follows:

$$\begin{aligned} ref_f(x^i_j) &= \{(i',j') \in ref(x^i_j) | i' < i \lor (i' = i \land j' < j)\}, \\ ref_b(x^i_j) &= \{(i',j') \in ref(x^i_j) | i' > i \lor (i' = i \land j' > j)\}, \end{aligned} \quad (5)$$

where $(i',j') \in ref(x^i_j)$ represents a reference from x^i_j to $x^{i'}_{j'}$. The ref_f and ref_b are used to decide the network structure of the encoder part in the proposed model.

3.2 Graph-based Encoder

In this section, we explain how to use the coreference relations in the encoder. Similar to the standard Seq2Seq model, the encoder of the proposed model is based on Bi-LSTM. For each input sentence $X_i = (x^i_1, \cdots)$, the forward encoder calculates the current hidden state $\overrightarrow{h}^i_t$ at the position of a word x^i_t as follows:

$$\overrightarrow{h}^i_t = \overrightarrow{LSTM}\left(emb(x^i_t), m(\overrightarrow{h}^i_{t-1}, ref_f(x^i_t))\right), \quad (6)$$

where $\overrightarrow{h}^i_{t-1}$ is the previous hidden state, $ref_f(x^i_t)$ is a set of words which are referred by x^i_t and $m(\cdot,\cdot)$ is a function which merges hidden state vectors. In this paper, we propose the following two functions as $m(\cdot,\cdot)$:

Coref-mean treats averaged hidden state vectors as the merged vector, as follows:

$$m(\overrightarrow{h}^i_{t-1}, ref_f(x^i_t)) = \frac{1}{|ref_f(x^i_t)| + 1}(\overrightarrow{h}^i_{t-1} + \sum_{(i',j') \in ref_f(x^i_t)} \overrightarrow{h}^{i'}_{j'}). \quad (7)$$

Coref-gate treats weighted sum of the hidden state vectors as the merged vector, as follows:

$$m(\overrightarrow{h}^i_{t-1}, ref_f(x^i_t)) = \overrightarrow{h}^i_{t-1} + \sum_{(i',j') \in ref_f(x^i_t)} \beta^{i'}_{j'} \odot \overrightarrow{h}^{i'}_{j'}, \quad (8)$$

where $\odot$ represents the element product for each dimension and $\beta^{i'}_{j'}$ represents the importance of $\overrightarrow{h}^{i'}_{j'}$. $\beta^{i'}_{j'}$ is calculated as follows:

$$\beta^{i'}_{j'} = sigmoid(W_t \overrightarrow{h}^{i'}_{j'} + W_s \overrightarrow{h}^i_{t-1}), \quad (9)$$

where W_t and W_s are weight matrices.

The backward encoding is similarly processed by replacing ref_f with ref_b. Finally, the forward and backward hidden states are concatenated to $h^i_t = [\overrightarrow{h}^i_t; \overleftarrow{h}^i_t]$ for each t. After that, h^i_t is used for translation, in place of h_t in equation (1), with attending only to the target sentence to be translated.

4 Experiments

4.1 Experimental Setting

We evaluated the proposed models on the English-to-Japanese translation data set in OpenSubtitles2018 (Lison et al., 2018). We cut out consecutive n $(= 1, 2, 3, 5, 7)$ sentences from the original data set as a unit. After that, we randomly selected 2000 units as test data, and the remaining about 1.87 million units were used as training data. All Japanese texts were tokenized by MeCab[3] with NEologd (Sato et al., 2017).

We set the vocabulary size for both source and target sides as 32,000. Both the encoder and the decoder were composed of 2-layer LSTMs. The dimension size of word embeddings for both source and target sides was set to 500. The dimension size of the encoder LSTM layers, the decoder LSTM layers, and an attention layer were set to 500, 1000, and 500, respectively. Initial values for weights were randomly sampled from a uniform distribution within the range of -1 to 1 (Glorot and Bengio, 2010).

Adam (Kingma and Ba, 2014) was used to update weight parameters, and the learning rate was set to 0.001. Learning was carried out for 200,000 steps for the entire training data. The mini-batch size was set to 32, and the gradients were averaged by the number of examples in each mini-batch. The order of mini-batches was randomly shuffled at the start of the training. Pytorch was used to implement the models. All models were run on a single GPU NVIDIA Tesla P100[4] independently.

We changed the number of input sentences, n, in the range of $\{1, 2, 3, 5, 7\}$ to observe the relationships between translation quality and the number of input sentences. We input a sentence to be

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

[4] This device has a 16GB memory.

	Number of sentences (n)				
	1	*2*	*3*	*5*	*7*
Cor-m	7.84	8.06	8.33	8.65	8.68
Cor-g	7.96	8.46	8.60	8.75	**8.79**
Cor-g-c	-	-	8.58	8.73	8.70
Concat	7.69	7.90	7.91	7.81	×

Table 1: BLEU scores for each model. The bold indicates the best score. The underlined indicates that these scores are statistically significantly improved from the score of the baseline Concat at the same setting ($p < 0.05$). × represents that the model did not run due to the shortage of GPU memories.

$n = 1$	$n = 2$	$n = 3$	$n = 5$	$n = 7$
12.8%	13.2%	13.8%	14.6%	15.4%

Table 2: The percentage of sentences containing coreferences in the test set.

translated and $n - 1$ sentences that precede the input sentence.

As a baseline model, we used a method concatenating multiple input sentences and generating a single sentence, proposed by Bawden et al. (2018) (**Concat**)[5]. We compared our proposed models, Coref-mean (**Cor-m**) and Coref-gate (**Cor-g**), with the baseline. In order to evaluate the effectiveness of succeeding sentences, we also experimented with the cases of inputting the same number of preceding and succeeding sentences for the target sentence to be translated at the center, for **Cor-g**. We denote this setting as Coref-gate-centered (**Cor-g-c**). The number of weight parameters for each model is 111,057k for the baseline and **Cor-m**, and 111,558k for **Cor-g**.

We used BLEU scores (Papineni et al., 2002) to evaluate the translation performance for each model. All reported BLEU scores in the experiments are averages for three times and are based on MeCab tokenization. Significance tests were conducted by paired bootstrap resampling (Koehn, 2004) with multevel (Clark et al., 2011)[6].

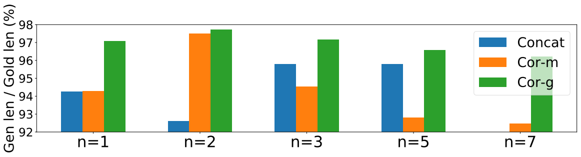

Figure 3: The ratio of token numbers in generated translations to those in reference translations.

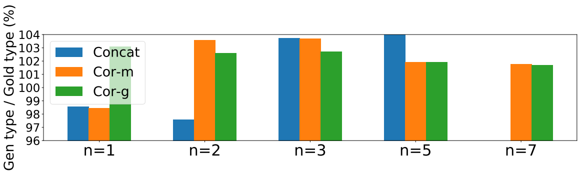

Figure 4: The ratio of token types in generated translations to those in reference translations.

4.2 Results and Analysis

Table 1 shows the results[7]. In this table, we can observe that our proposed models, **Cor-m** and **Cor-g**, outperformed the baseline **Concat** in terms of BLEU scores at every unit length. Interestingly, at the setting of $n = 1$, **Cor-g** also outperformed **Concat**. As shown in Table 2, this is because our proposed models can also use inter-sentential coreference information for translation. In the setting of $n = 2$, all the results improved from those for $n = 1$. This is consistent to the reported results in Bawden et al. (2018). In the setting of $n > 2$, improvement of BLEU scores for **Concat** stopped at $n = 3$, in contrast to the proposed models. This indicates that the proposed model can handle more sentences well by using their graph-based encoder and provided coreference information.

The scores for **Cor-g** is always better than those for **Cor-m**. From this result, we can say that the gating mechanism in **Cor-g** works well. In addition, as shown in Figure 3, the translation of **Cor-g** has a closer token length to the reference, while **Concat** and **Cor-m** encounter severe undergeneration problems. The results in Figure 4 show that in $n > 2$, **Cor-g** can maintain word coherence without increasing word types in generated sentences. Taking into account the gain of the BLEU scores, these results support our estimation that **Cor-g** can capture contexts well, compared to **Cor-m** and **Concat**.

However, the scores for **Cor-g-c** degraded compared to **Cor-g** at the same sentence numbers. This result reflects a tendency that most coreferences

[5] In our preliminary comparison, there are no statistically significant differences in translation performances between **Concat** and the method of inputting and outputting concatenated multiple sentences, also proposed by Bawden et al. (2018). From the computational efficiency perspective, therefore, we chose **Concat** as our baseline.

[6] https://github.com/jhclark/multeval

[7] These results are close to the reported BLEU scores of the Ja-En caption translations in Pryzant et al. (2018)

are anaphora, and cataphora is rarely observed in the test set. Ignoring the succeeding sentences, **Cor-g-c** at $n = 3, 5, 7$ is similar to the setting of **Cor-g** with $n = 2, 3, 4$. Interestingly, **Cor-g-c** at $n = 3, 5$ achieved better BLEU scores, compared to **Cor-g** with $n = 2, 3$. This indicates that cataphora information is also useful to translate many sentences in a text.

5 Conclusion

In this paper, we proposed a Seq2Seq model that can incorporate information in preceding and succeeding sentences of the translating sentence effectively, by taking into account provided coreference relations explicitly. Experimental results showed that the proposed models can improve the translation quality in the setting of inputting multiple sentences jointly, compared to the previous model. From these results, we could conclude that considering explicit coreference relations in the Seq2Seq model actually contributes to improve the performances on the English-to-Japanese translation.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv e-prints*, abs/1409.0473.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA. Association for Computational Linguistics.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. 2018. Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. European Language Resource Association.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. 2018. JESC: Japanese-English subtitle corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Toshinori Sato, Taiichi Hashimoto, and Manabu Okumura. 2017. Implementation of a word segmentation dictionary called mecab-ipadic-neologd and study on how to use it effectively for information retrieval (in japanese). In *NLP*, pages NLP2017–B6–1. The Association for Natural Language Processing.

Dario Stojanovski and Alexander Fraser. 2018. Coreference and coherence in neural machine translation: A study using oracle experiments. In *Proceedings of* *the Third Conference on Machine Translation: Research Papers*, pages 49–60, Belgium, Brussels. Association for Computational Linguistics.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274. Association for Computational Linguistics.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark. Association for Computational Linguistics.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium. Association for Computational Linguistics.