Neural Machine Translation Incorporating Named Entity

[1]Arata Ugawa [2]Akihiro Tamura [2]Takashi Ninomiya
[1,3]Hiroya Takamura [1]Manabu Okumura

[1]Department of Information and Communications Engineering, Tokyo Institute of Technology
[2]Graduate School of Science and Engineering, Ehime University
[3]National Institute of Advanced Industrial Science and Technology

ugawa.a.aa@m.titech.ac.jp
{tamura, ninomiya}@cs.ehime-u.ac.jp
{takamura, oku}@pi.titech.ac.jp

Abstract

This study proposes a new neural machine translation (NMT) model based on the encoder-decoder model that incorporates named entity (NE) tags of source-language sentences. Conventional NMT models have two problems enumerated as follows: (i) they tend to have difficulty in translating words with multiple meanings because of the high ambiguity, and (ii) these models' ability to translate compound words seems challenging because the encoder receives a word, a part of the compound word, at each time step. To alleviate these problems, the encoder of the proposed model encodes the input word on the basis of its NE tag at each time step, which could reduce the ambiguity of the input word. Furthermore, the encoder introduces a chunk-level LSTM layer over a word-level LSTM layer and hierarchically encodes a source-language sentence to capture a compound NE as a chunk on the basis of the NE tags. We evaluate the proposed model on an English-to-Japanese translation task with the ASPEC, and English-to-Bulgarian and English-to-Romanian translation tasks with the Europarl corpus. The evaluation results show that the proposed model achieves up to 3.11 point improvement in BLEU.

1 Introduction

Neural machine translation (NMT) models based on the encoder-decoder model, also known as the sequence-to-sequence model (Sutskever et al., 2014), have successfully shown their quality translation. Consequently, various NMT models are studied in the field of machine translation. To date, the most successful model is the bi-directional multi-layered encoder-decoder model with long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and an attention mechanism (Luong et al., 2015; Bahdanau et al., 2015), also known as attention-based NMT. LSTM and the attention mechanism are introduced to mitigate the difficulty in handling long sentences in the encoder-decoder model. The conventional attention-based NMT model is known to achieve high translation accuracy in bilingual evaluation understudy (BLEU). However, this model encounters two general problems: (i) it tends to have difficulty in translating words with multiple meanings because their translations have high ambiguity, and (ii) translation of compound words seems difficult because the encoder receives only a word, a part of the compound word, at each time step.

This study proposes a new NMT model based on the encoder-decoder model, incorporating named entity (NE) information in source-language sentences. The proposed model alleviates the problems of the conventional attention-based NMT by incorporating information of NE tags to the encoder and modeling chunk information of NE tags as the encoder's network structures. The encoder of the proposed model

Proceedings of the 27th International Conference on Computational Linguistics, pages 3240–3250
Santa Fe, New Mexico, USA, August 20-26, 2018.

encodes the input word based on its NE tag at each time step to reduce the ambiguity of the input word. Furthermore, the encoder introduces a chunk-level LSTM layer over a word-level LSTM layer and hierarchically encodes a source-language sentence to capture a compound NE as a chunk based on its NE tags. We evaluate the proposed model on an English-to-Japanese translation task with the Asian Scientific Paper Excerpt Corpus (ASPEC) (Nakazawa et al., 2016) and English-to-Bulgarian and English-to-Romanian translation tasks with the Europarl corpus. The evaluation results show that the proposed model achieves up to 3.11 point improvement in BLEU.

The main contributions of this study are as follows:

1. Semantic class information of NE tags is incorporated to the encoder of the attention-based NMT model.

2. Chunk information of NE tags is modeled as a part of network structures in the attention-based NMT model.

2 Related Work

This section describes the attention-based NMT model, a previous NMT model incorporating linguistic features of source-language sentences, and previous NMT models based on chunks/phrases.

2.1 Attention-based NMT Model

The attention-based NMT model is an NMT model, wherein an attention mechanism is introduced in the encoder-decoder model. The encoder-decoder model consists of two recurrent neural networks (RNNs), namely, an encoder and a decoder. Gated Recurrent Unit (Cho et al., 2014) or LSTM is typically used as units of RNNs. In this work, LSTM is employed as units of RNNs.

Given a word sequence of the source language $x = (x_1, x_2, \ldots, x_j, \ldots, x_{T_x})$, the encoder generates a hidden vector $(h_1, h_2, \ldots, h_j, \ldots, h_{T_x})$. Each hidden state of the encoder $h_j \in \mathbb{R}^{d \times 1}$ is calculated using LSTM, given the previous state $h_{j-1} \in \mathbb{R}^{d \times 1}$ and the input word x_j, as follows:

$$h_j = f_{enc}(h_{j-1}, E_x(x_j)), \tag{1}$$

where f_{enc} is the encoder's LSTM, E_x is an embedding layer, and $E_x(x_j)$ is an embedding vector of the word x_j.

Then, the decoder sequentially generates a word sequence of the target language $y = (y_1, y_2, \ldots, y_i, \ldots, y_{T_y})$, given the final state of the encoder, h_{T_x}. The decoder outputs a word sequence having the maximum logarithmic likelihood with respect to the input word sequence x as the output word sequence y:

$$\log p(y|x) = \sum_{i=1}^{T_y} \log p(y_i|y_{1:i-1}, h_{T_x}). \tag{2}$$

The probability distribution of the output word y_i is calculated from the hidden state $\bar{h_i}$ of the decoder:

$$p(y_i|y_{1:i-1}, h_{T_x}) = softmax(proj(\bar{h_i})), \tag{3}$$

where $proj$ is a projection function to resize a decoder's hidden state to a vector, wherein the vector's dimension is the target vocabulary size. $softmax$ is a softmax function to convert the projected vector o into a probability distribution by normalizing each element of o as follows:

$$o = proj(\bar{h_i}), \tag{4}$$

$$softmax(o) = \frac{\exp(o)}{\sum_{k=1}^{K} \exp(o_k)}, \tag{5}$$

where K is the vocabulary size of the target language.

To prevent deterioration of the translation performance for long source-language sentences, the attention-based NMT tries to capture the relation between the states in the encoder and the decoder, and the decoder generates the translations by referring to the history of the hidden states in the encoder. In particular, the alignment score a_i is calculated on the basis of the similarity between the decoder's hidden state $\bar{h_i}$ and each of the encoder's hidden states $c = (h_1, h_2, \ldots, h_j, \ldots, h_{T_x})$. The context vector s_i is calculated as the weighted average, where a weight is a_i, over all the encoder's hidden states:

$$a_i(j) = \frac{\exp(\bar{h_i} \cdot h_j)}{\sum_{j'=1}^{T_x} \exp(\bar{h_i} \cdot h_{j'})}, \tag{6}$$

$$s_i = \sum_{j=1}^{T_x} a_i(j) h_j, \tag{7}$$

where T_x is the length of the source-language sentence x, and $\cdot$ denotes the inner product. In the decoder, the probability distribution of the output word is calculated on the basis of the context vector s_i in addition to the decoder's hidden state $\bar{h_i}$:

$$p(y_i|y_{1:i-1}, c) = softmax(proj([\bar{h_i}; s_i])), \tag{8}$$

where [;] denotes the concatenation of the two vectors.

2.2 Linguistic Input Features for NMT

Sennrich and Haddow (2016) have improved the attention-based NMT model by using the linguistic features of source-language sentences. In particular, the encoder receives the lemma, subword tags, POS tags, and dependency labels of source-language sentences in addition to source-language words. A lemma is the original form of the word. Subword tags express prefix, stem, and suffix. A POS tag is part-of-speech information of a word, such as a noun or a verb. Moreover, a dependency label indicates a syntactic relation between words, such as a head and dependents. The encoder converts each of the linguistic features into its embedding vector, and then generates hidden states $(h_1, h_2, \ldots, h_j, \ldots, h_{T_x})$ from the word embedding vectors of source-language words $x = (x_1, x_2, \ldots, x_j, \ldots, x_{T_x})$ and the linguistic feature's embedding vectors as follows:

$$h_j = f_{enc}(h_{j-1}, (E_{word}(x_j)[\,;\, {}^{|F|}_{k=1} E_k(x_{jk})])), \tag{9}$$

where E_{word} is an embedding layer for source-language words, E_k is an embedding layer for the k-th type of linguistic features, and $|F|$ is the number of types of linguistic features (i.e., $|F|$ = 4). Note that the model has not used NE tags as a linguistic feature, and the incorporation of NE tags into NMT is still under exploration.

2.3 NMT Based on Chunk/Phrase Units

Ishiwatari et al. (2017) have improved the attention-based NMT by designing chunk-based decoders, each of which models global dependencies by a chunk-level decoder and local word dependencies by a word-level decoder. In their decoders, the chunk-level decoder first generates a chunk representation. Then, the word-level decoder predicts each target word from the chunk representation.

Wang et al. (2017) have improved attention-based NMT by integrating a phrase memory, which stores target phrases provided by a statistical machine translation (SMT) (Pal et al., 2010) model, to perform a phrase-by-phrase translation rather than a word-by-word translation. Their model dynamically selects a word or phrase to be output at each decoding step.

Meanwhile Ishiwatari et al. (2017) and Wang et al. (2017) have incorporated chunks identified by a chunker and a SMT model, respectively, our work focuses on chunk information of NE tags. Note that our proposed model can incorporate general chunk information, such as chunks found by a chunker. We will leave this aspect for future work.

3 NMT Incorporating NE Tags

In this section, we propose a new NMT model, incorporating NE tags of source-language sentences. By considering NE types of source-language words, we aim to improve translation performance for words with multiple meanings. Furthermore, by using chunk information of NE tags (e.g., IO or BIO information), we aim to improve translation performance for compound words.

The proposed model assumes that an NE tag is attached to each word in source-language sentences by using an NE tagger. The proposed model receives the sequence of NE tags along with the sequence of source-language words and generates each encoder's hidden state on the basis of not only source-language words but also NE tags. Furthermore, the proposed model introduces a chunk-level LSTM layer over a word-level LSTM layer into the encoder and hierarchically encodes a source-language sentence.

In Section 3.1, we overview NE tags, and in Section 3.2, we describe the model architecture of the proposed model.

3.1 Named Entity (NE)

NEs are words/phrases for specific entities, such as the name of persons, organizations, and locations. Moreover, NEs are sometimes extended to include time expressions and numerical representations. In CoNLL2003 shared task[1], the four types of NEs, namely 'person', 'organization', 'location', and 'names of miscellaneous', have been used. In Message Understanding Conference (MUC)[2], the seven kinds of NEs, namely 'person', 'location', 'organization', 'time', 'date', 'money', and 'percent', have been used. Information Retrieval and Extraction Exercise (IREX)[3] have used the eight kinds of NEs, where 'artifact' is added to the NEs used in MUC. Moreover, on Ontonotes 5.0[4], the 18 kinds of NEs have been defined. NEs could be considered as a kind of semantic category of words/phrases. Therefore, NEs have contributed to many NLP tasks, including SMT. Note that NEs have not been used in NMT.

NE Recognition is a classification task to identify NE words/phrases and their NE categories in given input sentences. In the tasks, NEs are usually expressed by chunk tags, such as BIO and IO tags, for words because an NE might be a phrase. For example, the sentence 'I arrived at Tokyo Station at 10:20.' is tagged by BIO2 tags as follows: 'I:O arrived:O at:O Tokyo:B-location Station:I-location at:O 10:20:B-time .:O', where B, I, and O indicate the beginning, inside, and outside of NEs, respectively. In this work, we used IO tags. By IO tags, NE words/phrases are represented as I (inside) and the others are expressed as O (outside). An example of the tagged sentence is as follows: 'I:O arrived:O at:O Tokyo:I-location Station:I-location at:O 10:20:I-time .:O'.

As mentioned above, NE tags include two features: (i) semantic class of words (e.g., location and time) and (ii) chunk information (e.g., I and O). By using the first feature, we aim to decrease the ambiguity of source-language words and improve the performance for words with high ambiguity. By using the second feature, we aim to capture the chunks of compound words and improve the performance for compound words.

3.2 Model Architecture

Figure 1 shows the encoder of the proposed model. The encoder of the proposed model is composed of three components, namely, the embedding layer, word-level LSTM layer, and chunk-level LSTM layer.

The encoder receives the sequence of NE tags, $tag = (tag_1, tag_2, \ldots, tag_i, \ldots, tag_{T_x})$, along with the sequence of source-language words, $x = (x_1, x_2, \ldots, x_i, \ldots, x_{T_x})$, where tag_i denotes the NE tag of x_i. At each time step i, the encoder generates the word embedding vector for the source-language word x_i through a word embedding layer E_x. Additionally, the embedding vector for the NE tag tag_i is generated through a tag embedding layer E_{tag}. Then, these vectors are added:

$$\hat{x}_i \quad = \quad E_x(x_i) + a * E_{tag}(tag_i), \tag{10}$$

[1] https://www.clips.uantwerpen.be/conll2003/

[2] http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_toc.html

[3] http://nlp.cs.nyu.edu/irex/

[4] https://catalog.ldc.upenn.edu/ldc2013t19

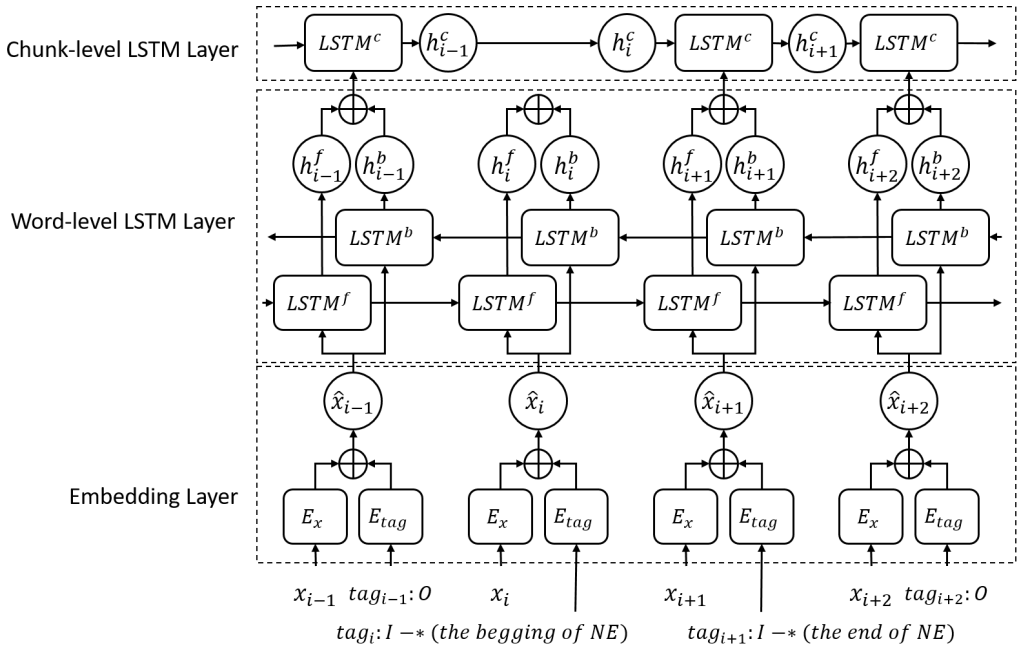

Figure 1: Encoder of the Proposed Model

where a is a parameter to control the influence of NE tags and is optimized in model training. The added vector, $\hat{x}_i$, is fed to a word-level LSTM component.

The word-level LSTM layer is a bi-directional LSTM, which encodes an input sentence by word. The bi-directional LSTM consists of a forward LSTM, $LSTM^f$, and a backward LSTM, $LSTM^b$. The forward LSTM obtains hidden states $(h^f_1, \ldots, h^f_{T_x})$ from the beginning to the end of the sentence:

$$h^f_i = LSTM^f(h^f_{i-1}, \hat{x}_i). \tag{11}$$

The backward LSTM obtains hidden states $(h^b_{T_x}, \ldots, h^b_1)$ from the end to the beginning of the sentence:

$$h^b_i = LSTM^b(h^b_{i+1}, \hat{x}_i). \tag{12}$$

Then, the bi-directional LSTM computes the average of h^f_i and h^b_i as the i-th word-level encoder's hidden state h_i:

$$h_i = (h^f_i + h^b_i)/2. \tag{13}$$

The hidden state h_i is fed to the chunk-level LSTM layer.

The chunk-level LSTM layer is a uni-directional forward LSTM, $LSTM^c$, which receives the hidden states of the word-level layer and encodes a source sentence by chunk identified by NE tags. Based on the chunk information of NE tags, the hidden states of the chunk-level LSTM layer, $(h^c_1, \ldots, h^c_{T_x})$, is calculated as follows:

$$h^c_i = \begin{cases} LSTM^c(h^c_{i-1}, h_i) & \text{if } tag_i \text{ is 'O' or the last part of the NE tags} \\ h^c_{i-1} & \text{otherwise} \end{cases} \tag{14}$$

The decoder of the proposed model is the same as in the conventional attention-based NMT model. The initial state of the decoder is set to the sum of the final state of the word-level LSTM layer and that of the chunk-level LSTM layer.

4 Experiment

This section compares the proposed model, which incorporates NE tags of source sentences, with a conventional NMT model, which does not use NE tags, under English-to-Japanese (En-Jp), English-to-Bulgarian (En-Bg), and English-to-Romanian (En-Ro) translation tasks to confirm the effectiveness of the proposed model.

Table 1: Statistics on Experimental Data (# of parallel sentences)

	Training Data	Development Data	Test Data
En-Jp	1,320,591	1,768	1,802
En-Bg	363,112	3,000	3,000
En-Ro	357,247	1,972	3,000

Table 2: Vocabulary Size

	Source Language	Target Language
En-Jp	78,591	57,771
En-Bg	27,872	30,000
En-Ro	27,651	30,000

4.1 Experimental Data

We evaluated the En-Jp translation performance on the ASPEC, which is used in WAT 2017[5], and the En-Bg and En-Ro translation performance on the Europarl corpus (Koehn, 2005).

The English and Japanese sentences are tokenized by spaCy[6] and KyTea (Neubig et al., 2011), respectively. The Bulgarian and Romanian sentences are tokenized by byte-pair encoding (Sennrich et al., 2016) implemented in sentencepiece[7], where we set the vocabulary size to 30,000. The words that appeared less than five times in the En-Jp training data and those less than twice in the English side of the En-Bg and En-Ro training data were replaced with the special symbol $\langle \text{UNK} \rangle$. In the training, all words were lowercased by lowercase.perl[8], and long sentences with over 50 words were filtered out.

In the En-Jp task, we used the development and test data employed in WAT 2017. In the En-Ro task, we used the newsdev-2016 as the development data and randomly sampled 3,000 parallel sentences from the training data as the test data. In the En-Bg task, we randomly sampled 3,000 parallel sentences from the training data for the development data and test data. The statistics on the experimental data are summarized in Table 1 and Table 2.

We used spaCy NE tagger[9], which was trained from the OntoNotes5.0[10] corpus attached with NE tags, to identify NE tags of English sentences in the proposed model. Table 3 shows the types of NE tags, and Table 4 gives the Top 3 NE types on the training corpus.

4.2 Competing Models

We compared the proposed model (*Proposed*) with the standard attention-based encoder-decoder NMT model, wherein the encoder is two stack bi-directional LSTM and the decoder is two stack forward LSTM (*Baseline*). Note that the proposed model introduces the embedding layer for NE tags and chunk-level forward LSTM layer into *Baseline*. We also compared the proposed model with the simulated model of Sennrich and Haddow (2016) (*Baseline_Concat*). In *Baseline_Concat*, the encoder generates a hidden state from the concatenated vector of the embedding vector for an input word and that for its NE tag at each time step (refer to Equation (9)). The network structures other than the embedding layer are the same as in *Baseline*.

All of the embedding size and hidden size of the encoder and decoder both in the baselines and proposed model are set to 256. The parameter a, which controls the effect of NE tags, is initialized to 0.5.

We used Adam (Kingma and Ba, 2015) with a mini-batch size of 64, 128, and 64 for learning each parameter in the En-Jp, En-BG, and En-Ro tasks, respectively. We also employed a gradient clipping

[5] http://orchid.kuee.kyoto-u.ac.jp/WAT/WAT2017/index.html
[6] https://spacy.io/
[7] https://github.com/google/sentencepiece
[8] http://www.statmt.org/moses/
[9] https://spacy.io/usage/linguistic-features#101
[10] https://catalog.ldc.upenn.edu/ldc2013t19

Table 3: Types of NE Tags

NE Type	Descriptions
PERSON	people, including fictional
NORP	nationalities or religious or political groups
FACILITY	buildings, airports, highways, bridges, etc.
ORG	companies, agencies, institutions, etc.
GPE	countries, cities, states
LOC	non-GPE locations, mountain ranges, bodies of water
PRODUCT	objects, vehicles, foods, etc. (not services.)
EVENT	named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	titles of books, songs, etc.
LAW	named documents made into laws
LANGUAGE	any named language
DATE	absolute or relative dates or periods
TIME	times smaller than a day
PERCENT	percentage, including '%'
MONEY	monetary values, including unit
QUANTITY	measurements, as of weight or distance
ORDINAL	'first', 'second', etc.
CARDINAL	numerals that do not fall under another type
OTHER	others

Table 4: Top 3 NE Types

Corpus	Top 3 NE Types
En-Jp	O: 94.1%, ORG: 1.65%, CARDINAL: 1.12%
En-Bg	O: 91.3%, ORG: 3.56%, DATE: 1.16%
En-Ro	O: 91.2%, ORG: 3.58%, DATE: 1.16%

Table 5: Experimental Results (BLEU score (%))

	Baseline	*Baseline_Concat*	*Proposed*
En-Jp	29.38	28.17	29.62
En-Bg	37.99	37.48	38.38
En-Ro	34.31	36.77	37.44

technique with the clipping value of 5.0, dropout with the dropout ratio of 0.2, and the weight decay with the coefficient of 1.0×10^{-6}. The training stopped after 30 epochs, and the model that achieved the best BLEU score on the development data, was used for testing.

4.3 Results

We evaluated the translation performances via the case-insensitive BLEU4 metric (Papineni et al., 2002), which is computed by multi-bleu.perl[11]. Table 5 shows the experimental results.

From Table 5, the proposed model is better than *Baseline* for all the translation tasks, which shows that the effectiveness of the incorporation of NE tags of source sentences into NMT. In addition, the proposed model outperforms *Baseline_Concat* for all the translation tasks. This observation indicates that the proposed model makes use of NE tags more effectively than the previous model (Sennrich and Haddow, 2016).

5 Discussion

This work aims to improve the translation performance for compound words and words with multiple meanings by using NE tags of source sentences. In this section, we discussed the effectiveness of the proposed model through actual examples. Table 6 shows the output examples of the baseline model, *Baseline*, and the proposed model for three English sentences.

The word 'first' has multiple meanings in Japanese (e.g., 最初の (beginning), 一つの (one of), 一位の (top)). (a) of Table 6 shows that the proposed model correctly translates the word 'first', whereas the baseline model does not. We guess the reason is that the proposed model could disambiguate the meanings of the word 'first' on the basis of the NE tag 'ORDINAL'.

From (b) in Table 6, the proposed model correctly translates the English phrase 'between 0.2 to 0.5' into the Japanese phrase '０．２～０．５', whereas the baseline model fails and results in 'over- translation' (i.e., the Japanese words '～０．０５' are generated twice). Probably, the reason is that the

Table 6: Output Examples of Baseline and Proposed Models

(a) Example1

Input	about	60	%	of	the	first	peak	showed	ca2	+	dependence	.
NE tags	I-PERCENT	I-PERCENT	I-PERCENT	O	O	I-ORDINAL	O	O	O	O	O	O

Reference: 最初のピークの約６割がｃａ２＋依存性を示した．
Baseline: １つのピークの約６０％はｃａ＋依存性を示した．
Proposed: 最初のピークの約６０％はｃａ２＋依存性を示した．

(b) Example 2

Input	quantum	yields	of	their	decomposition	are	between	0.2	to	0.5	,
	depending	on	their	substituents	.						
NE tags	I-ORG	O	O	O	O	O	I-CARDINAL	I-CARDINAL	I-CARDINAL	I-CARDINAL	O
	O	O	O	O	O						

Reference: それらの分解の量子収量は，それらの置換基に依存して，０．２～０．５である。
Baseline: 分解の量子収率は０．２～０．０５～０．０５になり，それらの⟨UNK⟩に依存して０．０５５．０５μｍである。
Proposed : この分解の量子収量は，それらの⟨UNK⟩に依存して０．２～０．５である。

(c) Example 3

Input	a	new	electromagnetic	coupling	structure	has	been	proposed	for	a	millimeter	wave	dr	-	vco	.
NE tags	O	O	O	O	O	O	O	O	O	O	I-CARDINAL	O	I-GPE	I-GPE	I-GPE	O

Reference: ミリ波ｄｒ-ｖｃｏ用の新しい電磁結合構造を提案した。
Baseline: ミリ波ｄｒ板のための新しい電磁結合構造を提案した。
Proposed: ミリ波ｄｒ-ｖｃｏの新しい電磁結合構造を提案した。

[11] http://www.statmt.org/moses/

Table 7: Experimental Results to Confirm the Effectiveness of Chunk Information

	Baseline	*Baseline_Chunk*	*Proposed_IO*	*Proposed*
En-Jp	29.38	28.61	29.51	29.62
En-Bg	37.99	37.55	37.93	38.38
En-Ro	34.31	35.48	35.78	37.44

proposed model could consider the four words as one phrase on the basis of the chunk information (I/O) of the NE tags.

Furthermore, as can be seen from (c) in Table 6, the compound word 'dr - vco' is correctly translated to 'd r - v c o ' by the proposed model, whereas it is wrongly translated by the baseline model. This finding also indicates that the proposed model is better for the translation of compound words. An interesting observation from (c) is that although the NE type of 'dr - vco' is wrong (i.e., the correct type is 'OTHER' although the attached type is 'GPE'), the proposed model translates the compound word correctly. This finding indicates that the chunk information of NE tags could be useful to NMT even if the NE type is not correctly identified.

The above examples indicate that both semantic class information of words (i.e., NE types) and chunk information (i.e., I/O) of NE tags could be helpful to NMT. To quantitatively confirm the pure effectiveness of chunk information of NE tags, we evaluated the performance of the proposed model that uses only chunk information of NE tags (*Proposed_IO*). In particular, *Proposed_IO* receives either of 'I' tag or 'O' tag as the NE tag. In addition, we evaluate the *Baseline* model that naively uses chunk information of NE tags (*Baseline_Chunk*). In *Baseline_Chunk*, each source sentence is preliminarily chunked on the basis of NE tags and each chunk is treated as a word. For example, the two words 'donald:I-PERSON trump:I-PERSON' are concatenated into one chunk 'donald_trump' and the chunk is treated as one word.

Table 7 summarizes their performance. As can be seen from Table 7, *Baseline_Chunk* is worse than *Baseline* in the En-Jp and En-Bg tasks although *Baseline_Chunk* outperforms *Baseline* in the En-Ro task. This finding indicates that the naive incorporation of chunk information of NE tags could have negative effect to NMT. Table 7 also shows that *Proposed_IO* is at least comparable to, or better than, *Baseline*. This observation indicates that only chunk information of NE tags could improve NMT performance in the proposed model. We conjecture that chunking increases vocabulary sizes. Thus, the naive incorporation (i.e., *Baseline_Chunk*) might suffer from the data sparseness problem, whereas *Proposed_IO* might alleviate the sparseness problem by the hierarchical structure of the encoder (i.e., word-level LSTM layer). In addition, Table 7 shows that *Proposed* outperforms *Proposed_IO* for all the tasks. This finding indicates that NE types enable further improvement in NMT performance.

6 Conclusion

We have proposed a new encoder-decoder NMT model, which alleviates the problems of word ambiguities and compound words in the conventional attention-based NMT by incorporating NE tags of source-language sentences. The encoder of the proposed model encodes both input word and NE tag at each time step and has a chunk-level LSTM layer over a word-level LSTM layer to hierarchically encode a source-language sentence.

The experiments on the English-to-Japanese translation task with the ASPEC show that the proposed model achieved 0.24 point improvement in BLEU. In addition, the experiments on the English-to-Bulgarian and English-to-Romanian translation tasks with the Europarl corpus show that the proposed model achieved 0.39 and 3.11 point improvements in BLEU, respectively.

We qualitatively analyzed translation results to investigate the effectiveness of the NE information in the encoder-decoder model and found several examples that show the effectiveness of either semantic class information or chunk information of NE tags.

Finally, we also conducted experiments to evaluate how chunk information (i.e., I/O) of NE tags contributed to improve the translation accuracy without semantic class information. From the experiments, we observed that chunk information could improve the translation accuracy, and the translation accuracy

was further improved by adding semantic information of NE tags.

Future work includes finding good models to incorporate NE tags and verify the effectiveness of the proposed models for other datasets. In addition, we would like to incorporate an NE tagging model to the proposed model to learn both the NE model and machine translation model.

Acknowledgments

This work was partially supported by JSPS Grants-in-Aid for Scientific Research Grant Number 25280084. We appreciate Hidetaka Kamigaito for his helpful suggestions on this work. We also thank the anonymous reviewers for their careful reading of our study and their insightful comments.

References

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations, arXiv:1409.0473*.

Kyunghyun Cho, Bart van Merrienboer, aglar Gulehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 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*, pages 1724–1734.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. In *Proceedings of The Neural computation*, volume 9, pages 1735–1780. MIT Press.

Shonosuke Ishiwatari, Jingtao Yao, Shujie Liu, Mu Li, Ming Zhou, Naoki Yoshinaga, Masaru Kitsuregawa, and Weijia Jia. 2017. Chunk-based decoder for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1901–1912.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations, arXiv:1412.6980*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the MT summit*, pages 79–86.

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.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the 10th Conference on International Language Resources and Evaluation*, pages 2204–2208.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 529–533.

Santanu Pal, Sudip Kumar Naskar, Pavel Pecina, Sivaji Bandyopadhyay, and Andy Way. 2010. Handling named entities and compound verbs in phrase-based statistical machine translation. In *Proceedings of the 2010 Workshop on Multiword Expressions: from Theory to Applications*, pages 46–54.

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

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Trunslation*, volume 1, pages 83–91.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of The Advances in neural information processing systems*, pages 3104–3112.

Xing Wang, Zhaopeng Tu, Deyi Xiong, and Min Zhang. 2017. Translating phrases in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1421–1431.