IEICE TRANS. INF. & SYST., VOL.E103–D, NO.4 APRIL 2020

PAPER

Korean-Vietnamese Neural Machine Translation with Named Entity Recognition and Part-of-Speech Tags

Van-Hai VU[†], ***Student Member***, **Quang-Phuoc NGUYEN**[†a)], **Kiem-Hieu NGUYEN**[††], **Joon-Choul SHIN**[†], ***and*** **Cheol-Young OCK**[†b)], ***Nonmembers***

SUMMARY Since deep learning was introduced, a series of achievements has been published in the field of automatic machine translation (MT). However, Korean-Vietnamese MT systems face many challenges because of a lack of data, multiple meanings of individual words, and grammatical diversity that depends on context. Therefore, the quality of Korean-Vietnamese MT systems is still sub-optimal. This paper discusses a method for applying Named Entity Recognition (NER) and Part-of-Speech (POS) tagging to Vietnamese sentences to improve the performance of Korean-Vietnamese MT systems. In terms of implementation, we used a tool to tag NER and POS in Vietnamese sentences. In addition, we had access to a Korean-Vietnamese parallel corpus with more than 450K paired sentences from our previous research paper. The experimental results indicate that tagging NER and POS in Vietnamese sentences can improve the quality of Korean-Vietnamese Neural MT (NMT) in terms of the Bi-Lingual Evaluation Understudy (BLEU) and Translation Error Rate (TER) score. On average, our MT system improved by 1.21 BLEU points or 2.33 TER scores after applying both NER and POS tagging to the Vietnamese corpus. Due to the structural features of language, the MT systems in the Korean to Vietnamese direction always give better BLEU and TER results than translation machines in the reverse direction.

key words: *Korean-Vietnamese machine translation, neural machine translation, named entity recognition, part-of-speech, word sense disambiguation*

1. Introduction

Recently, the relationship between South Korea and Vietnam has improved significantly in various fields, from the economy, to politics, to culture. These new connections have led to a need for MT to improve the convenience of information exchange. However, research papers about Korean-Vietnamese MT remains rare. In this paper, we extend our previous research [1] by applying NER and POS to the Vietnamese corpus as a pre-processing step to improve the performance of Korean-Vietnamese NMT systems.

We implemented POS and NER tagging by merging the Bi-directional Long Short Term Memory (Bi-LSTM) with a Conditional Random Field (CRF). After passing the Bi-LSTM, a source sentence is transformed into a matrix that becomes the input for the CRF, which forecasts which named entity or POS that will best correspond to the input sentence. During POS and NER tagging of the Vietnamese corpus, the text and its categories (a kind of NER or categories of POS) are added. In the following stage, the NMT system embeds all the source text in a continuous vector before using a Recurrent Neural Network (RNN) to encode the source sentence into a sequence of word vectors (word embedding). Then, the NMT decodes those embedded words to predict the target sentence.

In Vietnamese, many words have different meanings or diverse grammatical categories depending on their context. For example, the word "*vinh*" can be the name of a person or the name of a city. In another example, the word "*tiến*" is both a noun indicating a person's name and a verb with *progressive* meaning. Applying our NER and POS tools, removes the semantic and grammatical ambiguity from the Vietnamese corpus, giving the MT more parameters with which to analyze input data and predict target sentences. We also use the UTagger tool for Korean text to generate a corpus with Word Sense Disambiguation (WSD).

This paper conducted a series of experiments on Korean-Vietnamese bi-directional translation systems using our corpus of more than 450K sentence pairs. The implementation results show that the quality of the Korean-Vietnamese translation is enhanced by NER and POS tagging. The most significant improvement of 1.39 BLEU points or 2.38 TER scores were exhibited in a Korean-Vietnamese MT system that used both NER and POS.

2. Related Works

2.1 Korean-Vietnamese Machine Translation Systems

Nguyen et al. [2] used UWordMap to establish a Korean lexical semantic network in which each sense of every polysemous word is connected to a sense-code that constitutes a network node. After tagging the Korean corpus using UWordMap, Nguyen performed Korean-Vietnamese translation experiments with OpenNMT [3]. However, that research used a small parallel corpus (281K sentence pairs) for training.

In another study, Cho et al. [4] used a statistical MT (SMT) phrase table to create a morpho-syntactic filter for solving the problem of the lexical gap. Depending on the lexical choice, they grouped component morphemes of Korean adjectives and Korean verbs. After they used the Moses toolkit for training data and translation from

Manuscript received June 4, 2019.
Manuscript revised September 27, 2019.
Manuscript publicized January 15, 2020.

[†]The authors are with University of Ulsan, Ulsan, Republic of Korea.
[††]The author is with Hanoi University of Science and Technology, Hanoi, Vietnam.
a) E-mail: nqphuoc@gmail.com (Corresponding author)
b) E-mail: okcy@ulsan.ac.kr (Corresponding author)

DOI: 10.1587/transinf.2019EDP7154

Korean to Vietnamese, they used the BLEU and TER score to evaluate the performance of their translation. The translation quality improved by approximately one point in BLUE scores, and that decreased by over two point in TER scores.

Nguyen et al. [5] also indicated that the translation quality improved when analyzed morphologies were used to train a parallel corpus. They built a Korean-Vietnamese MT system using the Moses toolkit. Before training the MT model with 24K sentence pairs, they analyzed the morphologies of the Korean text. The experimental results showed that analyzing the morphologies improved the MT quality by 3.34 BLEU points.

Cho et al. [6] built a Korean-Vietnamese SMT MT based on the Moses toolkit. They selected data, including words, phrases, and sentences inside brackets, quotation marks, and parentheses, that were translated individually. This simple method was effective with the training data for a sentence containing brackets, quotation marks, or parentheses.

Most previous Korean-Vietnamese MT systems, including those just described, are based on SMT. However, several research papers [7], [8] have indicated that NMT gives better results than SMT.

In this paper, we use NMT for training and translation from Korean to Vietnamese and vice versa. First, however, we apply NER and POS to the Vietnamese text in the parallel training corpus.

2.2 Machine Translation Using POS and NER

Several researchers have applied POS or NER to MT. Ueffing et al. [9] applied POS information to an English-Spanish/Catalan SMT. Before they trained their MT model, Ueffing et al. tagged the English text with POS. The BLEU and Word Error Rate scores showed that applying POS in the SMT significantly improved the performance of the machine.

Blinkov et al. [10] evaluated the quality of NMT with POS tagging. Some of their experiments translated Arabic into English/Hebrew and French/Czech into English. They used the BLEU score to evaluate the NMT performance and found that translation into a morphologically rich language (Hebrew/German) is more difficult than translation into a morphologically poor language (English).

A paper published by Niehued et al. [11] also applied POS and NER to NMT. Their experiments translated German into English, and their system improved the translation quality by 1.5 BLEU points.

Balabantaray [12] showed that NER positively affects MT quality. Based on association rules and assumptions, they demonstrated a newly named entity class recognition method. After they tagged the English corpus with 19 different types of entities, they used an English-Odia parallel corpus for their MT system.

Bhalla et al. [13] improved the quality of MT using NER. Their work used proper names, location names, organization names, and miscellaneous as entities that they classified using a statistical method. The Moses toolkit was used to build an SMT system from English to Punjabi with a parallel corpus of about 50K sentence pairs.

Most previous research has focused on English as a source language. Our research applied POS and NER to the Korean-Vietnamese parallel corpus.

2.3 Vietnamese POS and NER

Several papers have mentioned POS or NER for Vietnamese. Le [14] proposed improving the accuracy of a NER system for Vietnamese by combining regular expressions over tokens with a bidirectional inference method in a sequence labeling model. Then, he used the overall F_1 score for accurate evaluation, and the result was 89.66%.

Le [15] demonstrated an NER system for a Vietnamese corpus using a label propagation algorithm. They presented three methods for labeling documents (choosing noun phrases as named entity candidates, measuring word similarity, and decreasing the effect of high-frequency labels). Their experimental results indicated improvements over their previous system.

Our previous research combined LSTMs with CRFs in a neural architecture for labeling [16]. The experiments showed remarkable results, with F1 scores of 93.52% and 94.88% for POS and NER, respectively.

3. Tagging POS and NER Using Bi-Directional LSTM-CRFs

POS and NER are challenging sequence-labeling problems. A POS defines a category of words that have similar grammar (verbs, nouns, adjective, etc.), and NER identifies entities (persons, locations, organizations, etc.). Most traditional sequence label tagging has used supervised learning techniques such as CRFs [17], hidden Markov models [18], or maximum entropy Markov models [19]. However, model sequence label tagging has used neural network architectures instead, because of their effectiveness [20]–[23]. Here, we explain the structure of a recurrent neuron and present the background of our Bi-LSTM (an improvement on RNNs) and how we apply CRFs for POS and NER tagging.

3.1 Recurrent Neuron

Figure 1 shows the recurrent neuron, in which one neuron receives an input, generates an output and then sends the output back to itself. At time step t, each neuron N receives both input x_t and the output from the previous step y_{t-1}. The output y_t is computed by

$$y_t = f\left(x_t^T W_x + y_{t-1}^T W_y + b\right), \tag{1}$$

where W_x and W_y are two matrix weights for the input x_t and the output of the previous time step y_{t-1}; f is the active function; and b is a vector of n neurons containing each neuron's bias term.

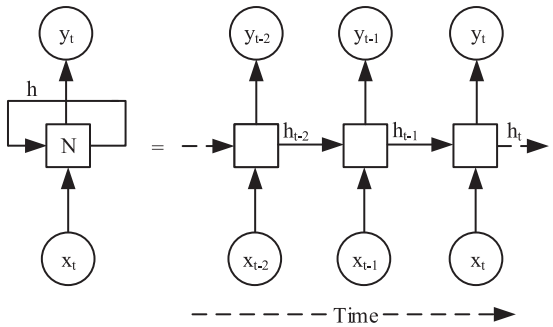

Fig. 1 Structure of a recurrent neuron.

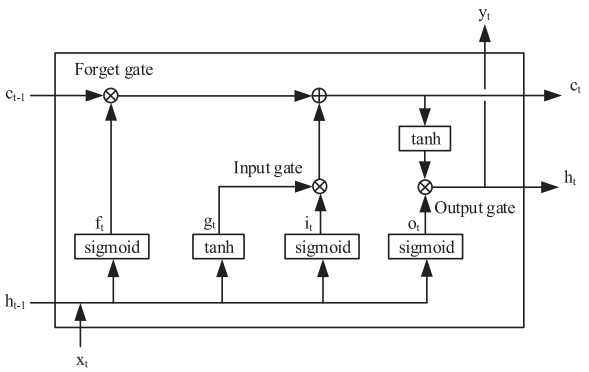

Fig. 2 An LSTM cell: "c" stands for the cell, and the hidden shorts are written as "h."

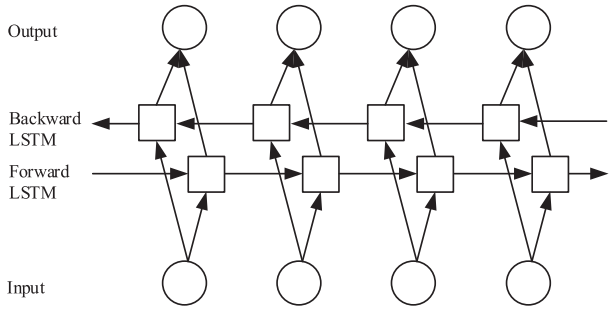

Fig. 3 Structure of Bi-LSTM.

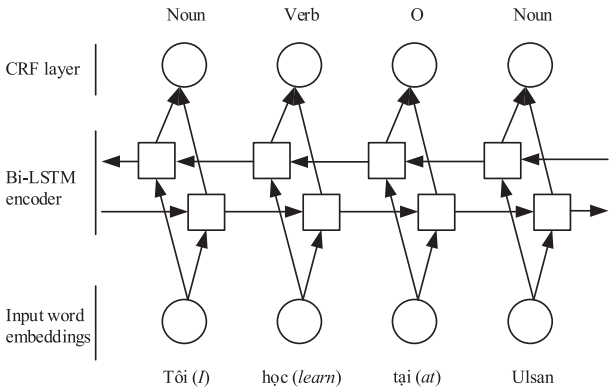

Fig. 4 Bi-LSTM-CRFs for POS.

The h_t shortcut for the hidden state at state t is calculated based on the current input and the hidden state at the previous time step as:

$$h_t = f(h_{t-1}, x_t). \tag{2}$$

3.2 Bidirectional LSTM

LSTM was introduced by Hochreiter & Schmidhuber [24], and its special ability is learning from long-term dependencies. Every RNN has the form of a series of repetitive modules, and each recurrent cell includes just a tanh layer. However, each LSTM cell uses four layers (three sigmoid layers and one tanh layer) that interact with each other as:

At step t, the input gate i_t, forget gate f_t, output gate o_t, output of main layer g_t, cell vector c_t, hidden layer h_t, and output layer y_t are respectively computed by

$$f_t = \sigma\left(W_{xf}^T x_t + W_{hf}^T h_{t-1} + b_f\right) \tag{3}$$

$$g_t = tanh\left(W_{xg}^T x_t + W_{hg}^T h_{t-1} + b_g\right) \tag{4}$$

$$i_t = \sigma\left(W_{xi}^T x_t + W_{hi}^T h_{t-1} + b_i\right) \tag{5}$$

$$o_t = \sigma\left(W_{xo}^T x_t + W_{ho}^T h_{t-1} + b_o\right) \tag{6}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes g_t \tag{7}$$

$$y_t = h_t = o_t \otimes \tanh(c_t) \tag{8}$$

where W_{xf}, W_{xg}, W_{xi}, and W_{xo} are the weight matrices of the four layers connected to the input x_t; W_{hf}, W_{hg}, W_{hi}, and W_{ho} are the weight matrices of the four layers connected to the previous hidden state h_{t-1}; and b_f, b_g, b_i, and b_o are the bias terms of the latter four layers.

In the Bi-LSTM model, the output at step t depends on both the front elements and the behind elements. For example, to forecast missing words in a sentence, it is necessary to consider both the previous parts and the next parts of the sentence. Therefore, we can consider the model as the overlap of two LSTM networks facing each other. At this time, the output is calculated based on the hidden states of both LSTM networks. The Bi-LSTM structure is shown in the following figure:

3.3 Bi-LSTM-CRFs for POS and NER Tagging

Figure 4 illustrates the structure of Bi-LSTM-CRFs for POS tagging. CRFs are a probability model often applied to the predictive structures in sample identity and machine learning. In the combination of Bi-LSTM and CRF, a sequence input that has passed the Bi-LSTM becomes the input for the CRF layer. Then, the CRF layer predicts the named entity output sequence that best corresponds to the input string.

The Bi-LSTM-CRF structure for NER is similar to that for POS. In the system of Nguyen [16], word embedding for POS includes word2vec and character representation, whereas the input for NER is the concatenation of word2vec, character representation, chunk, and POS. The output of a Bi-LSTM-CRF for POS is the text with POS

Table 1 POS tagging performance on VietTreebank dataset.

Method	Accuracy	Evaluation Method
NNVLP	91.92	5-fold cross-validation
RDRPOStagger	92.59	
Bi-LSTM-CRFs	**92.98**	
VNTagger	93.40	10-fold cross-validation
Bi-LSTM-CRFs	**93.52**	

Table 2 NER tagging performance on Vietnamese language and speech processing 2016 dataset.

Method	P	R	F1
Bi-LSTM-CRFs	90.97	87.52	89.21
Bi-LSTM-CRFs + POS	90.90	90.39	90.64
Bi-LSTM-CRFs + Chunk	95.24	92.16	93.67
Bi-LSTM-CRFs + POS + Chunk	**95.44**	**94.33**	**94.88**

tagging, and the output of a Bi-LSTM-CRF for NER is a sentence in which the name of the entity corresponds to each word.

Compared with previous POS and NER tagging tools, our tool [16] showed remarkable improvements, as shown in Table 1 and Table 2. In our experiments, we used cross-validation methods to assess the quality of the POS tagging tool, and we used the micro-averaged F1 score to evaluate the performance of the NER tagging program.

4. Neural Machine Translation

NMT integrates neural language models and a traditional statistical MT system into one system. In the 1990s, the research of Castano [23] and Forcada [24] showed the use of neural networks to train a translation model. However, because of hardware limitations, NMT was abandoned for almost two decades.

Hardware advances have allowed the performance of NMT to be improved significantly. In 2014, Sutskever et al. [27] and Cho et al. [28] proposed a sequence-to-sequence framework for NMT models. After using an RNN to encode the input sequence into a fixed length vector representation, their method used another RNN to decode the target sequence from that vector.

This paper presents a neural translation model based on a sequence-to-sequence encoder-decoder with the attention that we used to establish our NMT system.

4.1 Encoder

The encoder is a bi-directional RNN, including a forwarding RNN and a backward RNN, as shown in Fig. 5. The first task of the encoder is to transform the input sentence into a sequence of word vectors (word embedding). Then, the encoder processes those vectors using a bi-directional RNN.

Mathematically, the source sentence is a sequence of the form $x = (x_1, x_2, \ldots, x_n)$, where n is the length of the sentence. The hidden states of the forwarding RNN ($\overrightarrow{h}$) and backward RNN ($\overleftarrow{h}$) are calculated by

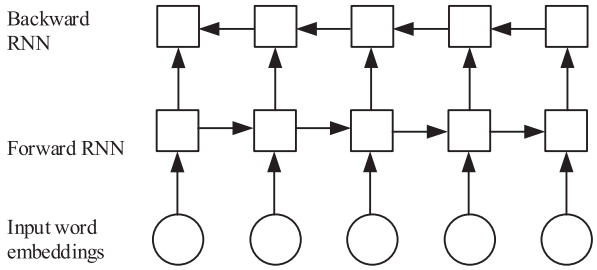

Fig. 5 Encoder structure.

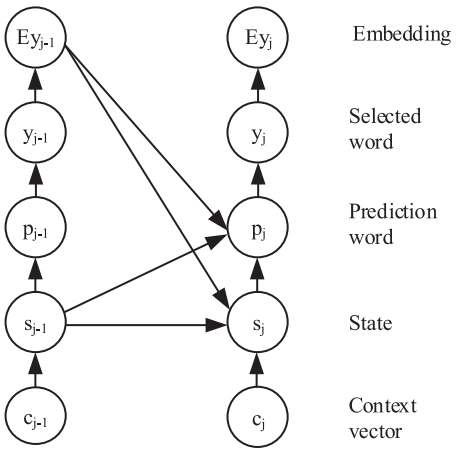

Fig. 6 Decoder structure.

$$\overrightarrow{h_i} = f\left(\overrightarrow{h_{i-1}}, \bar{E}x_i\right) \tag{9}$$

$$\overleftarrow{h_i} = f\left(\overleftarrow{h_{i+1}}, \bar{E}x_i\right) \tag{10}$$

In Eqs. (9), and (10), f is a tanh function (a typical feed-forward neural network layer) - $f(x) = \tanh(Ax + B)$. $\bar{E}$ is a word-embedding matrix of the source language.

The source annotations $(h_1, h_2, \ldots, h_n)$ are a concatenation of the forward and backward hidden states as:

$$h_i = \left(\overrightarrow{h_i}, \overleftarrow{h_i}\right). \tag{11}$$

4.2 Decoder

The decoder, shown in Fig. 6, is a forwarding RNN used to predict the target sentence $y = (y_1, y_2, \ldots, y_m)$, where m is the length of the sentence. A sequence of the hidden state s_j is computed from the previously hidden state s_{j-1}, the previous target word Ey_{j-1}, and the input context vector c_j using the following equation:

$$s_j = f(s_{j-1}, Ey_{j-1}, c_j). \tag{12}$$

The prediction vector p_j is based on the input context c_j, the decoder hidden state s_{j-1}, and the embedding of the previous output word Ey_{j-1}.

$$p_j = softmax(W(Us_{j-1} + Ey_{j-1} + Cc_j)) \tag{13}$$

In Eqs. (12), and (13), E is the word-embedding matrix

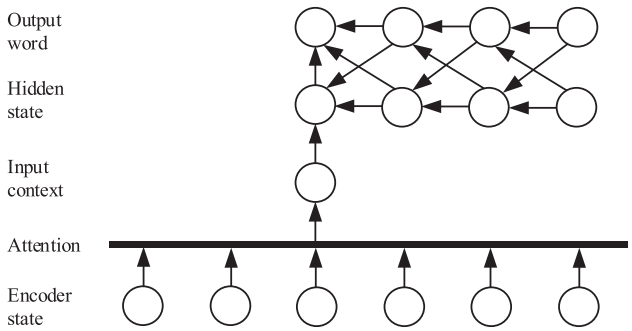

Fig. 7 Attention mechanism structure.

of the target language, and W, U, and C are weight matrices.

The output word y_j will be selected to have the highest value in p_j before using its embedded Ep_j for the following stage.

4.3 Attention Mechanism

The attention mechanism is a combination of input word representation $\left(\overrightarrow{h_i}, \overleftarrow{h_i}\right)$ (which is generated in the encoder step) and the context state c_j (produced from the previously hidden state of the decoder s_{j-1}). The structure of the attention mechanism is visualized in Fig. 7.

The attention value is computed as:

$$e_{ij} = \frac{exp(a(s_{j-1}, h_i))}{\sum_k exp(a(s_{j-1}, h_k))}, \tag{14}$$

where $\boldsymbol{a}$ is the association between the decoder state and each input word. This association is calculated by:

$$a(s_{j-1}, h_i) = W^{aT} s_{j-1} + U^{aT} h_i + b^a, \tag{15}$$

where $\boldsymbol{W^a}$ and $\boldsymbol{U^a}$ are weight vectors, and $\boldsymbol{b^a}$ is the bias value.

At that time, the value of the context vector c_j is computed by the following equation:

$$c_j = \sum_i e_{ij} h_i. \tag{16}$$

5. Experiments and Results

This reseach carried out a series of bi-directional translation experiments between Korean and Vietnamese to assess the effects of NER and POS on the performance of our NMT.

5.1 Datasets

When conducting experiments, we used our Korean-Vietnamese corpus, which has more than 454K sentence pairs [1], to train the MT models. This corpus is the state-of-the-art and greatest parallel corpus for Vietnamese-Korean translation systems.

Table 3 shows the number of sentences, average sentence length, tokens, and vocabulary of our Korean-Vietnamese corpus. In the Vietnamese corpus, we applied a segmented sentence tool [29] for word segmentation (Word Seg.), which decreased the average sentence length and tokens from 19.3 and around 8.7M to 16.3 and just over 7.4M, respectively. Those numbers remained unchanged after using POS and NER [16].

Table 3 The Korean-Vietnamese parallel corpus.

		#Sentences	#Avg. Length	#Tokens	#Words
Vietnamese	Original	454,751	19.3	8,790,197	40,090
	Word Seg.		16.3	7,409,163	49,208
Korean	Original		12	5,435,686	397,130
	MA + WSD		21.4	9,728,801	68,856

Table 4 Vietnamese sentences after applying Word Seg., NER, and POS.

Form	Sentences
Original	tôi đang làm việc tại hàn quốc.
Word Seg.	tôi đang làm_việc tại hàn_quốc.
Word Seg. + POS	tôi|PN đang làm_việc |V tại hàn_quốc|N.
Word Seg. + NER	tôi__B-PER đang làm_việc tại hàn_quốc__B-LOC.
Word Seg. + POS + NER	tôi|PN__B-PER đang làm_việc |V tại hàn_quốc|N__B-LOC.

In Korean corpus, UTagger was used to generate new sentences with a morphological analysis (MA) and WSD, which increased the average sentence length and number of tokens from 12 to 21.4 and more than 5.4M to more than 9.7M, respectively.

Whereas the segmented sentence tool created a new corpus with more words than the original corpus, the MA and WSD generated a new corpus with fewer words than the original corpus. More details are given in our previous paper [1].

5.2 Implementation

Before putting the corpora into our NMT, we carried out many pre-processing steps. In the Vietnamese sentences, we first used the tool of Nguyen [29] to segment words. The task of Word Seg. is dividing the written text into meaningful units, as shown in Table 4. Then, we applied our tool [16] for POS and NER tagging. The method of applying POS and NER to the Vietnamese corpus changed the form of the sentences, as shown in Table 4. The sentence in the table means "*I am working in Korea*," in which the word "*tôi*" is transformed into "*tôi|PN*" and "*tôi__B-PER*" after applying POS and NER, respectively. PN (pronoun), N (noun), and V (verb) are the tagged POS. B-PER (begin person) and B-LOC (begin location) are the tagged NER.

In the Korean corpus, we used UTagger for MA and WSD. Table 5 describes the transformation of a basic Korean sentence after applying UTagger. This sentence means "*I am working in Korea*," but the word "*il*" can have several different meanings: *work*, *day*, or *one*. WSD transformed those words into "*il__01*," "*il__02*," or "*il__03*," to

Table 5 Korean sentences transformed after applying UTagger.

Form	Sentences
Original	(na neun han-kuk e-seo il ko iss seup-ni-da .)
UTagger	(na_03 neun han-kuk_05 e-seo il__01 ko iss_01 seup-ni-da .)

Table 6 Translation results in BLEU and TER points.

Systems		BLEU	TER
Korean-to-Vietnamese	Baseline	25.64	65.10
	UTagger	27.79	58.77
	UTagger + NER	28.41	56.94
	UTagger + POS	28.94	56.63
	UTagger + NER+POS	**29.18**	**56.39**
Vietnamese-to-Korean	Baseline	12.88	70.61
	UTagger	25.44	58.72
	UTagger + NER	25.92	57.07
	UTagger + POS	26.38	56.56
	UTagger + NER+POS	**26.47**	**56.44**

reflect those meanings. In this case, "*il*" is transformed into "*il__01*" with the meaning "*work*."

To evaluate the effects of NER and POS on individual translation qualities, we established the five systems based on deep learning model (sequence-to-sequence with attention model). The parallel corpus for training in different systems were described as below:

• *Baseline:* Uses the original Korean and Vietnamese texts with Word Seg. by RDRsegmenter.

• *UTagger:* Uses Vietnamese sentences from the baseline MT system and the Korean corpus modified by UTagger (Korean MA and WSD).

• *UTagger + NER:* Korean sentences from the UTagger MT system and Vietnamese from the baseline MT system pre-treated with NER tagging.

• *UTagger + POS:* Korean sentences from the UTagger MT system and Vietnamese from the baseline MT system pre-treated with POS tagging.

• *UTagger + NER + POS:* Korean sentences from the UTagger MT system and Vietnamese from the baseline MT system pre-treated with both NER and POS tagging.

The NMT systems were implemented in the OpenNMT framework [3]. The parameters for training were set with 2x500 RNNs as the hidden layer, the word-embedding dimension as 500, and the input feed as 13 epochs. We randomly selected 2000 sentence pairs for testing and used the rest for training.

5.3 Results

The BLEU [30] and TER [31] have been used in this research to evaluate the translation performance of our bi-directional Korean-Vietnamese MT systems. Table 6 shows the quality of eight NMT systems in terms of BLEU and TER scores.

In general, the BLEU scores in the Korean-to-Vietnamese direction are higher than those in the reverse direction. The number of tokens in the Korean sentences is higher than the number in the Vietnamese sentences (approximately 10 times before using UTagger and more than 1.7 times after using UTagger for Korean sentences), as shown in Table 3. The BLEU score evaluates the performance of an MT system based on the probability (P) of the result sentence and the test sentence (*P (the result sentence/the test sentence)*). Each word in the test sentence is an extract from the total vocabulary. A large amount of vocabulary thus leads to a small P, lowering the BLEU score.

As shown in Table 6, applying UTagger to the Korean sentences significantly increased the BLEU score of the Vietnamese-to-Korean MT system from 12.88 points to 25.44 points because of the significant reduction in vocabulary that results from applying UTagger in Table 3. See, for instance, the following test set of paired sentences:

• *Vietnamese: "tôi đi đến trường học." (I go to the school.)*

• *Korean*: *"na_neun hag_gyo_e gan_da."*

The MT system translated the Vietnamese sentence to *"na_nul hag_gyo_e ga_da*," which is correct in terms of meaning, but the BLEU sees that *"gan_da"* and *"ga_da"* are different, which lowers BLEU score. After using UTagger on the Korean sentences, the above example sentence in the test set and the translation sentence are transformed into *"na nul hag gyo e gan da."* and *"na nul hag gyo e ga da."* The BLEU sees only that "*gan*" differs from "*ga*," which increases the BLEU score. The number of words is decreased significantly after using UTagger. Therefore, the BLEU score for the Vietnamese-Korean MT systems improves significantly when using UTagger.

5.3.1 Effect of Vietnamese Named Entity Recognition

In Vietnamese, various words have different meanings depending on their context. Our Vietnamese NER system indicates whether a word is the name of a person, a location, an organization, or none of the above. For instance: the word "*Huế*" can be the name of a person (tagged *Huế|B-PER*) or a city (tagged *Huế|B-LOC*). NER thus clarifies the input text, which increases the accuracy of the MT systems. Table 6 shows that after applying NER to Vietnamese texts, the quality of the Korean-Vietnamese and Vietnamese-Korean MT systems increased by 0.62 points and 0.48 points, respectively, compared with the baseline systems.

The use of NER for Vietnamese also reduced the TER score by 1.83 points in the Korean-to-Vietnamese NMT system and by 1.65 points in the reverse direction.

5.3.2 Effects of Vietnamese Parts-of-Speech

Similar to the case of NER, our tool tags POS using 20 categories. For instance, the word "*đại*" can be a person's name (tagged as *đại|PN*) or an adjective that means big (tagged as *đại|Adj*), depending on the context. The POS tags thus indicate a word's syntactic function, allowing the MT systems to more easily find words with the same meaning and function in the target language. As a result, the quality of the MT systems is improved. Specifically, our Korean-Vietnamese NMT system increased its BLEU score by 1.15 points, and

the Vietnamese-Korean NMY improved its score by 0.94 points compared with the baseline systems.

In the Korean-Vietnamese MT systems, applying POS produced a BLEU score of 28.94 points, 0.63 points higher than the MT after applying NER. In the Vietnamese-Korean MT, the performance after applying POS was 0.46 points greater after applying NER. POS produces greater gains than NER because the number of categories in POS tagging is much higher than in NER tagging (20 kinds of POS tagging compared with 6 labels for NER tagging).

In term of TER evaluation, the translation error rate was reduced from 58.77 to 56.63 in Korean-to-Vietnamese MT and from 58.72 to 56.56 in Vietnamese-to-Korean MT after applying POS for Vietnamese.

5.3.3 Effect of Combining NER and POS in Vietnamese Sentences on Translation Quality

The combination of both NER and POS makes the input data clearer, as shown in Table 4. The simultaneous use of POS and NER solved the problems of multiple meanings and grammatical diversity in Vietnamese. Compared with the MT without NER and POS (UTagger MT system), the Korean-Vietnamese MT system and the Vietnamese-Korean MT system increased their performance by 1.39 and 1.03 BLEU points, respectively, as shown in the following table.

The quality of the translation is assessed by the TER scores, which also shows that the combination of NER and POS for Vietnamese have the highest efficiency with an average error reduction rate of 2.33 points in the Korean-Vietnamese MT system and the reverse direction system.

6. Conclusion

In this paper, we applied NER and POS to Vietnamese sentences in a Korean-Vietnamese corpus inherited from our previous research paper. Then we built bi-directional Korean-Vietnamese NMT systems and compared their results with previous results. The BLEU and TER scores demonstrate that NER and POS positively affects the bi-directional Korean-Vietnamese MT. The improvement in the Korean-Vietnamese translation direction is more significant than in the reverse direction in all paired NMT systems.

In the future, we intend to apply syntactical dependency to both Korean and Vietnamese sentences to further improve the performance of our MT systems.

Acknowledgments

This work was supported by the Ministry of Education of the Republic of Korean and the National Research Foundation of Korea ((NRF-2019S1A5B6102698)) and the ICT R&D Program of MSIP/IITP (Developmentof Core Technology for Context-aware Deep-Symbolic Hybrid Learning and Construction of Language Resources) under Grant 2013-0-00179.

References

[1] Q.-P. Nguyen, A.-D. Vo, J.-C. Shin, P. Tran, and C.-Y. Ock, "Building a Korean-Vietnamese neural machine translation system with Korean morphological analysis and word sense disambiguation," IEEE Access, pp.1–13, 2019.

[2] Q.-P. Nguyen, A.-D. Vo, J.-C. Shin, and C.-Y. Ock, "Neural Machine Translation Enhancements through Lexical Semantic Network," Proc. 10th International Conference on Computer Modeling and Simulation - ICCMS 2018, Sydney, Australia, pp.105–109, 2018.

[3] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "OpenNMT: Open-Source Toolkit for Neural Machine Translation," Proc. 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, Canada, pp.67–72, July 2017.

[4] S.W. Cho, E.-H. Lee, and J.-H. Lee, "Phrase-Level Grouping for Lexical Gap Resolution in Korean-Vietnamese SMT," in Computational Linguistics, vol.781, K. Hasida and W.P. Pa, eds. Springer Singapore, Singapore, pp.127–136, 2018.

[5] Q.-P. Nguyen, J.-C. Shin, and C.-Y. Ock, "Korean morphological analysis for Korean-Vietnamese statistical machine translation," J. Electron. Sci. Technol., vol.5, no.4, pp.413–419, Dec. 2017.

[6] S.-W. Cho, Y.-G. Kim, H.-S. Kwon, E.-H. Lee, W.-K. Lee, H.-M. Cho, and J.-H. Lee, "Embedded clause extraction and restoration for the performance enhancement in Korean-Vietnamese statistical machine translation," Proc. 28th Annual Conference on Human & Cognitive Language Technology, Busan, Korea, pp.280–284, 2016.

[7] L. Bentivogli, A. Bisazza, M. Cettolo, and M. Federico, "Neural versus Phrase-Based Machine Translation Quality: a Case Study," Proc. 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, pp.257–267, 2016.

[8] M. Junczys-Dowmunt, T. Dwojak, and H. Hoang, "Is neural machine translation ready for deployment? A case study on 30 translation directions," arXiv:1610.01108 [cs], Oct. 2016.

[9] N. Ueffing and H. Ney, "Using POS information for statistical machine translation into morphologically rich languages," Proc. 10th conference on European chapter of the Association for Computational Linguistics, Budapest, Hungary, vol.1, pp.347–354, 2003.

[10] Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, and J. Glass, "What do Neural Machine Translation Models Learn about Morphology?," Proc. 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, Canada, pp.861–872, July 2017.

[11] J. Niehues and E. Cho, "Exploiting linguistic resources for neural machine translation using multi-task learning," arXiv:170800993 Cs, Aug. 2017.

[12] R.C. Balabantaray, "Name entity recognition in machine translation," Emerg. Technol., vol.1, no.3, p.3, 2010.

[13] D. Bhalla, N. Joshi, and I. Mathur, "Improving the quality of MT output using novel name entity translation scheme," Proc. 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, India, pp.1548–1553, 2013.

[14] H.-P. Le, "Vietnamese named entity recognition using token regular expressions and bidirectional inference," arXiv:161005652 Cs, Oct. 2016.

[15] H.T. Le, R.C. Sam, H.C. Nguyen, and T.T. Nguyen, "Named entity recognition in vietnamese text using label propagation," Proc. 2013 International Conference on Soft Computing and Pattern Recognition (SoCPaR), Hanoi, Vietnam, pp.366–370, Dec. 2013.

[16] A.-D. Nguyen, K.-H. Nguyen, and V.-V. Ngo, "Neural sequence labeling for Vietnamese POS tagging and NER," arXiv:181103754 Cs, Nov. 2018.

[17] R. Chopra, N. Singh, Y. Zhenning, and N.Ch.S.N. Iyengar, "Sequence Labeling using Conditional Random Fields," Int. J. U- E-Serv. Sci. Technol., vol.10, no.9, pp.101–108, Sept. 2017.

[18] A. Krogh, "Hidden Markov models for labeled sequences," Proc. 12th IAPR International Conference on Pattern Recognition (Cat. No.94CH3440-5), Jerusalem, Israel, vol.2, pp.140–144, 1994.
[19] P. Blunsom, "Maximum entropy Markov models for semantic role labelling," Proc. Australasian Language Technology Workshop 2004, Sydney, Australia, pp.109–116, Dec. 2004.
[20] W. Gunawan, D. Suhartono, F. Purnomo, and A. Ongko, "Named-Entity Recognition for Indonesian Language using Bidirectional LSTM-CNNs," Procedia Comput. Sci., vol.135, pp.425–432, 2018.
[21] J. Zhou and W. Xu, "End-to-end learning of semantic role labeling using recurrent neural networks," Proc. 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, pp.1127–1137, 2015.
[22] C.N. dos Santos and V. Guimarães, "Boosting named entity recognition with neural character embeddings," arXiv:150505008 Cs, May 2015.
[23] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural Architectures for Named Entity Recognition," Proc. 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, pp.260–270, June 2016.
[24] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Comput., vol.9, no.8, pp.1735–1780, Nov. 1997.
[25] M.A. Castaño, F. Casacuberta, and E. Vidal, "Machine translation using neural networks and finite-state models," Proc. 7th International Conference on Theoretical and Methodological Issues in Machine Translation, Santa Fe, USA, pp.160–167, 1997.
[26] M.L. Forcada and R.P. Ñeco, "Recursive hetero-associative memories for translation," Proc. Biological and Artificial Computation: From Neuroscience to Technology, Berlin, Heidelberg, vol.1240, pp.453–462, 1997.
[27] I. Sutskever, O. Vinyals, and Q.V. Le, "Sequence to sequence learning with neural networks," Proc. 27th Advances in Neural Information Processing Systems, Montreal, Canada, pp.3104–3112, 2014.
[28] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," Proc. 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp.1724–1734, Oct. 2014.
[29] D.Q. Nguyen, D.Q. Nguyen, T. Vu, M. Dras, and M. Johnson, "A fast and accurate Vietnamese word segmenter," Proc. 7th International Conference on Language Resources and Evaluation, Miyazaki, Japan, p.6, May 2018.
[30] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," Proc. 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania, pp.311–318, 2001.
[31] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," Proc. Association for Machine Translation in the Americas, p.9, 2006.

Van-Hai Vu received his B.S. from Hanoi University of Industry, Hanoi, Vietnam, in 2011 and his M.S. from VNU University of Engineering and Technology, a member of Vietnam National University, Hanoi, Vietnam, in 2014, both in information technology. Currently, he is a Ph.D. candidate at the University of Ulsan, Ulsan, Republic of Korea. His research interests include natural language processing, machine learning, and machine translation.

Quang-Phuoc Nguyen received his B.S. from the University of Sciences, Vietnam National University, Ho Chi Minh City, Vietnam, in 2005, his M.S. from Konkuk University, Seoul, Republic of Korea, in 2010, and his Ph.D. from the University of Ulsan, Ulsan, Republic of Korea both in information technology. His research interests include natural language processing, machine learning, and machine translation.

Kiem-Hieu Nguyen received his Ph.D. in information technology from the University of Ulsan, Korea, in 2013. From 2013 to 2015, he pursued postdoc research at LIMSI, CNRS, and CEA-LIST in France. He has been a lecturer/researcher at Hanoi University of Science and Technology, Vietnam, since 2015. His research interests include natural language processing and information extraction. He has publications at top-tier NLP conferences, such as ACL, COLING, CICLING, and LREC. He serves as a reviewer for several venues, including LREC, IJICAI, PAKDD, and IEEE Access. He has tight collaborations with industry (Samsung, VCCorp, VNPay) doing R&D for NLP and AI-powered, data-driven tasks.

Joon-Choul Shin received his B.S., M.Sc., and Ph.D. in information technology from the University of Ulsan in 2007, 2009, and 2014, respectively. Currently, he works as a postdoctoral researcher at the University of Ulsan. His research interests include Korean language processing, document clustering, and software engineering.

Cheol-Young Ock received his B.S., M.Sc., and Ph.D. in information technology from Seoul National University, Seoul, Republic of Korea, in 1982, 1984, and 1993, respectively. He was a visiting professor at TOMSK Institute, Russia, in 1994 and at Glasgow University, Glasgow, UK, in 1996. Currently, he is a professor at the University of Ulsan. His research interests include natural language processing, ontology, information retrieval, and machine learning.