Journal of Advanced Marine Engineering and Technology, Vol. 44, No. 1, pp. 68~74, 2020
J. Advanced Marine Engineering and Technology (JAMET)
https://doi.org/10.5916/jamet.2020.44.1.68

ISSN 2234-7925 (Print)
ISSN 2234-8352 (Online)
Original Paper

Syllable-based Korean named entity recognition using convolutional neural network

Yeon-Soo You[1] · Hyuk-Ro Park[†]

(Received December 17, 2019 ; Revised December 31, 2019 ; Accepted January 17, 2020)

Abstract: Named entity recognition (NER) finds object names in documents or sentences and classifies them into given categories. However, recognizing entity names in natural-language sentences is challenging, as it requires an understanding of various contexts. Recently, many researches have tried to apply deep learning methods to NER and have improved performance. Particularly, the bidirectional long-short-term-memory with conditional random field (bi-LSTM-CRF), which is a recurrent neural network model, is considered the most accurate for NER, as it considers contexts of both directions and is not affected by gradient-vanishing. However, the sequential nature of bi-LSTM-CRF makes the model extremely slow in training and classifying. To overcome this issue of computational speed, we propose a syllable-based Korean NER method using a convolutional neural network with CRF (CNN-CRF). The experiment with three corpora shows that the proposed model achieves a similar level of performance with bi-LSTM-CRF (0.4% improvement); however, it is 27.5% faster than bi-LSTM-CRF.

Keywords: Named entity recognition; Deep learning; Bi-LSTM-CRFs; CNN-CRFs

1. Introduction

A named entity refers to a proper noun indicating a certain object. Named entity recognition (NER) seeks expressions indicating names of objects in a document or sentence and classifies them into pre-defined categories. NER is used in various natural language processing and understating systems such as information search and extraction, automatic abstraction, question-answering and so on. People can easily understand contexts to extract and classify named entities; however, machines have difficulty in processing them because of the following problems.

A typical problem of NER is ambiguity. For example, a named entity “Seoul” can be used as a geographic area called “Seoul City” and an organization called “Seoul City Hall” depending on context. Another typical problem is processing of out-of-vocabulary (OOV) words; these words are newly created and not registered in the dictionary.

The methods of recognizing named entities change according to the language being used. In English documents, there are many cases of named entities starting with a capital letter (e.g., “Korea” and “China”) or a word “the” (e.g., “the Nile” and “the Sahara Desert”). However, unlike English NER, it is not easy to find distinct features (such as a capital letter or “the”) in the case of Korean NER (e.g. “한국” (hankook: Korea), “중국” (jungkook: China), “나일강” (nile-gang: Nile River), and “사하라 사막” (sahara-samak: Sahara Desert).

There are two methods of Korean NER: a method of recognizing a named entity in a morpheme unit based on morphological analysis; a method of recognizing a named entity in a syllable unit. The morphological-analysis-based method divides an input character string in morpheme units and uses Hangeul (Korean alphabet) part-of-speech (POS) as features **[1]-[4]**. Recently, many studies have been investigating methods to improve the performance of NER by adding features such as POS, pre-analyzed dictionary, and syllable-embedding **[2][5]-[7]**. The morphological-analyzer-based NER method has an advantage that features such as POS and dictionary can be used. However, erroneous analysis may occur in the case of proper nouns not being found in the dictionary of morphemic analyzer, and such erroneous information can affect the performance of the named entity

† Corresponding Author (ORCID: http://orcid.org/0000-0002-5594-6675): Professor, Department of Electronics & Computer Engineering, Chonnam National University, 77, Yongbong-ro, Buk-gu, Gwangju, 61186, Korea, E-mail: hyukro@jnu.ac.kr, Tel: +82-62-530-3431

1 M. S. Candidate, Department of Electronics & Computer Engineering, Chonnam National University, E-mail: powinz00@naver.com

recognizer, which is considered a drawback.

The syllable-based NER method divides a sentence in a sequence of syllables and uses them in NER **[8]-[11]**. Unlike the morphological-analysis-based method, features such as POS and pre-analyzed dictionary are not used. As no dictionary is being used, there is a possibility that the boundary of a word can be misanalyzed. In the case of a syllable-based NER for "서울시청" (seoul-si-cheong: Seoul City Hall), it can be misinterpreted as "서울시" (seoul-si: Seoul City) if the end of the named entity is analyzed incorrectly.

Nevertheless, syllable-based NER has an advantage that no resource is used other than the input sentence. As the morphological analyzer is not being used, there are no problems involving the propagation of errors during morphological analysis. Furthermore, a similar model can be used for various languages, in addition to Korean. Therefore, this paper studies a syllable-based NER method.

This paper is structured as follows: Section 2 discusses related works, and Section 3 describes the proposed model. Section 4 includes a comparative analysis of experimental results between a recurrent neural network (RNN) model and the proposed model. Finally, Section 5 provides the conclusion.

2. Related Work

2.1 Deep Learning and NER

The first step of NER is to convert a sequence of syllables (or morphemes) into a sequence of numbers in order to process using a computer. A unique number is assigned to every syllable (or morpheme) that appears in the training data. And all syllables (or morphemes) that appear in validation or test data but not appear in training data is given a specific number that denote an OOV syllable (or morpheme).

The numbers that denote a syllable or a morpheme is transformed into vectors so that they can be used in deep learning models. The vectors are computed in a way that they represent various aspects of the meanings of syllables or morphemes. For such a word-embedding vector, there are two methods: using pre-learned values and learning concurrently while training the NER model.

Data expressed using vectors are fed into a neural network such as RNN or convolutional neural network(CNN) and the network compute the probability of each named entity category for current input. The tag of the category with highest computed probability is assigned to the current input.

2.2 Korean NER

In the past, Korean NER used methods such as Hidden Markov Model (statistics-based machine learning), conditional random fields (CRFs), and structural support vector machine (SVM). Nowadays, however, deep-learning-based methods are mainly studied. The performance of early studies on deep-learning-based NER was similar to the that of conventional statistics-based methods (through a model using feed-forward neural network) **[12]**.

Another study **[13]** proposed a method using RNN that is suitable for sequential data.

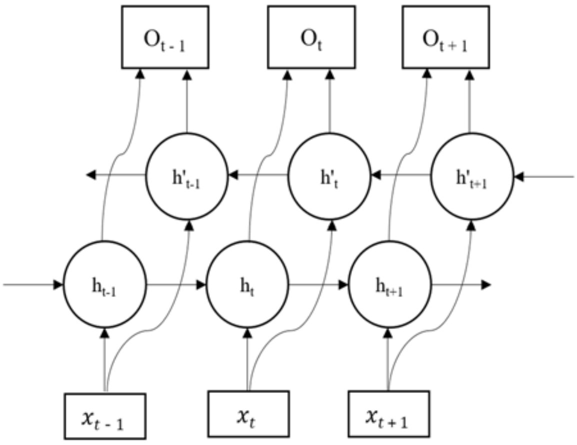

Figure 1: Bidirectional Recurrent Neural Network

Figure 1 shows a bidirectional RNN. For the forward calculation, the output of the previous hidden state is used to compute the output of the current hidden state.

$$h_t = \tanh\,(W_h h_{t-1} + W_x x_t + b) \tag{1}$$

$$O_t = h_t \oplus h'_t \tag{2}$$

As shown in **Equation (1)**, the current input x_t is multiplied by a weight matrix W_x and the previous hidden state output h_{t-1} is multiplied by another weight matrix W_h. These two values and a bias value b is added together and resulting value is inputted to the hyperbolic tangent function to produce h_t. For the backward computation, using the same method, h'_t is computed, and the values of both directions are concatenated to output O_t of **Equation (2)**.

RNNs can learn sequential dependencies in data; however, there exists a vanishing gradient issue, in which the learning is not properly performed if the length of the input data increases. Lately, long short-term memory (LSTM) that adds gates to RNN, is used to resolve the vanishing gradient problem.

However, in pure LSTM-based methods, only the context proceeding the current input is considered. To utilize both the proceeding and following contexts of current input, bi-directional LSTM is devised **[1]-[3][7][11][14][15]**. Furthermore, some studies added CRFs to the bi-LSTM models to reflect dependencies between output tags **[1]-[9][11][13]-[15]**.

In addition, a method of using stacked bi-LSTM that stacks multiple layers of bidirectional LSTM was proposed along with an ensemble model, which learns multiple models and chooses a value having the highest probability among the output values **[4][5][15]**.

For English NER, it is reported that the CNN-CRFs models perform inferior to bi-LSTM-CRFs **[14]**. However, no study has compared bi-LSTM-CRFs and CNN-CRFs for Korean NER.

In terms of Korean NER, RNN models are dominant models that use deep-learning. As RNN processes input data sequentially, parallel processing is impossible. Therefore, as the length of sentence increases, the processing speed slows down.

3. Proposed Model

This section introduces the CNN of deep learning and describes the proposed model using this.

3.1 CNN

Feed-forward neural network has a drawback that local features of input data cannot be extracted. CNN, which is usually used in image processing, is a method of extracting local features of data by performing convolution operations.

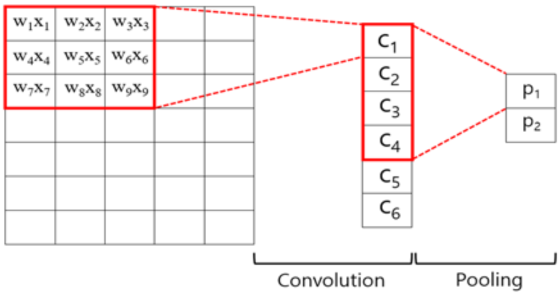

Figure 2: Convolutional Neural Network

As shown in **Figure 2**, a partial calculation is performed using a filter for input data. Local features can be extracted by dividing the whole data into areas of the same size with the filter and performing the following computation.

$$C_i = \text{ReLu}\left(\sum_{j=0}^{h} w_j x_j + b\right) \tag{3}$$

As shown in **Equation (3)**, an element x_i of an area with size h in input data and an element w_j of the filter are multiplied and all values are added. Using this sum and an activation function ReLu, output c_i is obtained.

$$\text{Relu}(\text{x}) = \max\,(0, x) \tag{4}$$

As shown in **Equation (4)**, the ReLu function outputs 0 for a value less than or equal to 0 and as-is for a value greater than 0.

All output c_i are fed into a pooling layer and the layer produce an output p_k. A pooling layer may be a max pooling or an averaging pooling layer. Max pooling uses the maximum value among the values of c_i as p_k, and average pooling uses an average value.

In general, CNN uses multiple filters of varying sizes to extract various features from input data.

The size of data decreases through a convolution operation because the convolution operation transforms an area of data to a single number using the filter multiplication. To make the size of the output the same as that of the input, we can add paddings which consists of all 0 values to the original data.

Furthermore, 2D or 3D convolution can be used, which can process 2D or 3D data depending on the type of input data. This CNN method is commonly used in image processing.

3.2 Proposed Model

A CNN-based model is proposed to improve the processing speed of the RNN model which is usually used in Korean NER. The proposed CNN-CRF model is shown in **Figure 3**. We put a limit to the length of input sentences. Before embedding, we make the lengths of all sentences same by adding paddings to the sentences having shorter length than the maximum length. Sentence-embedding uses a lookup table to convert a sentence into vectors that can properly represent the feature of each syllable. The lookup table is initialized with random numbers and as the learning is performed, the best-fitting values are learned.

Embedded data are inputted in the convolutional layer to extract features. 1D convolution is used for a sentence because calculation is performed in only one direction, including all embedding vectors of a syllable.

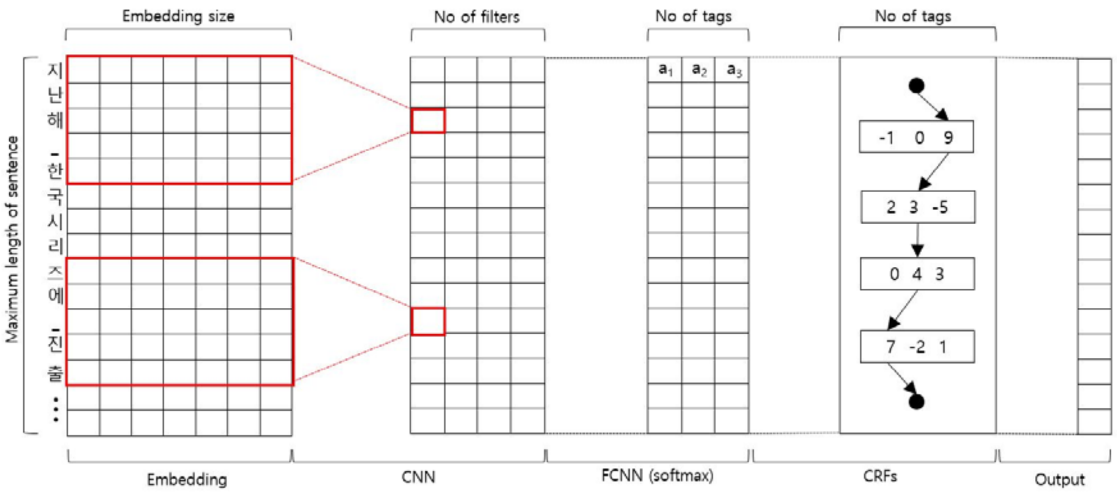

Figure 3: CNN-CRFs model

When a CNN is used, several types of filter are usually used. However, the proposed model extracts a feature by using only one type of filter having a size 5 to include two syllables, before and after current position.

To make the length of the output data the same as the maximum length of a sentence, two padding symbols are added at the beginning and end of the sentence. ReLu is used as the activation function, and the pooling layer is not used (as the input and output sizes are identical).

The output of the convolution layer is fed into a fully connected neural network which produce the probabilities of all categories for each input symbol. The shape of the output of the fully connected neural network is the length of the sentence times the number of categories. The output of the fully connected layer is converted to probability values between 0 and 1 using the Softmax function.

$$y_k = \frac{\exp(a_k)}{\sum_{i=1}^{n} \exp(a_i)} \tag{5}$$

Softmax is as shown in **Equation (5)**. The exponential function of an input value a_k is divided by the sum of exponential function of all input values, thereby outputting y_k. As the sum of all output y_k is the same as 1, this value can be used as the probability value.

This output value is the given to the CRFs, and a tag sequence corresponding to the whole sentence is finally obtained as the output. The CRFs are trained by considering the forward and backward dependency of the output tags, and the Viterbi algorithm is used to output the optimal tag sequence.

4. Experiments

4.1 Experimental Corpora

Experiments were conducted using NER corpora of Electronics and Telecommunication Research Institute (ETRI), Korean Information Processing System Competition (KIPSC) **[16]**, and Korea Maritime and Ocean University (KMOU) **[17]**. The corpora were converted in syllable units, and the maximum length was set to 180 to conduct the experiments in as similar environments as possible. To tag named entities, BIO tags were used to mark the beginning of named entity as B (Begin), the middle as I (Inside), and a syllable that is not a part of a named entity was marked as O (Outside).

Table 1 shows the details of each corpus. The ETRI's corpus (A) and the KMOU's corpus (C) consist of 10 categories: personal names (PER), location names (LOC), organization names (ORG), and others (POH); date (DAT), time (TIM), and duration (DUR) for time expressions; currency (MNY), percentage (PNT), and other numeral expressions (NOH) for numeral expressions.

Table 1: Corpora detail

	A	B	C
PER	16,387	7,147	55,138
ORG	15,831	14,252	68,604
LOC	5,805	5,628	27,478
DAT	4,372	9,195	23,704
TIM	651	1,519	2,796
POH	17,420		53,164
NOH	14,566		48,772
DUR	1,750		7,188
PNT	975		7,119
MNY	812		8,948
O	217,701	189,133	1,170,845
No. of sentences	4,637	3,517	23,749

The corpus distributed at the KIPSC (B) consists of five categories: personal names (PS), location names (LC), organization name (OG), date (DT), and time (TI). The corpus distributed at the KIPSC has the least number of sentences and contains relatively small number of personal names and large number of time expressions. Next, the corpus of ETRI consists of similar volumes with KIPSC corpus. The most recently distributed corpus of KMOU has the largest volume.

The experiments were performed by dividing each corpus into training dataset (80%), validation dataset (10%), and test dataset (10%).

4.2 Implemented Environment

Table 2 shows the equipment used in the experiments.

Each model was implemented using Tensorflow, an open-source deep learning framework provided by Google. And the five models were compared for each corpus. The word-embedding size of models was 128 and the dropout rate was 0.2, which were all identical. The other parameters are shown in **Table 3**.

Table 2: Equipment Specifications

Equipment	Specification
CPU	Intel® Xeon® CPU E5-2630 v4
RAM	32GB
GPU	Geforce GTX 1080

Table 3: Model parameters

Model	hidden units	filter size	Filters
bi-LSTM	64	-	-
bi-LSTM-CRFs	64	-	-
CNN	-	5	64
CNN-CRFs	-	5	64
CNN-CRFs-2	-	5	256

4.3 Experimental Results

Table 4 shows the experimental results. In the experiment using ETRI corpus, the performance of the bi-LSTM model was 0.5% better than the bi-LSTM-CRFs model. The bi-LSTM-CRFs (83.4%) performed better (83.4%) with the validation data than with the test dataset (82.7%). We believe the reason as the amount of learning data being insufficient. The corpus distributed by KIPSC showed a low performance overall. While analyzing the prediction results, the prediction results of KIPSC corpus showed a performance of approximately 90% for the O tag whereas those of the other corpuses showed a performance of more than 95% for the O tag. The models trained using the KIPSC corpus could not distinguish named entities and non-named entities properly because the volume of corpus was small.

Table 4: Experimental results by model

	Model	precision	recall	F1-score	Time
ETRI	bi-LSTM	84.3	82.4	**83.2**	**0.192 s**
	bi-LSTM-CRFs	85.3	80.7	82.7	0.562 s
	CNN	78.5	74.7	76.3	0.016 s
	CNN-CRFs	87.0	84.6	**85.6**	**0.359 s**
	CNN-CRFs-2	84.7	81.7	83.0	0.377 s
KIPSC	bi-LSTM	76.3	71.2	73.5	0.188 s
	bi-LSTM-CRFs	75.3	72.9	**73.9**	**0.517 s**
	CNN	78.7	71.8	74.8	0.016 s
	CNN-CRFs	84.2	75.3	79.0	0.384 s
	CNN-CRFs-2	83.1	76.6	**79.4**	**0.39 s**
KMOU	bi-LSTM	84.4	80.9	82.4	0.207 s
	bi-LSTM-CRFs	85.5	81.3	**83.2**	**0.575 s**
	CNN	77.1	65.2	69.2	0.013 s
	CNN-CRFs	86.5	80.0	83.0	0.385 s
	CNN-CRFs-2	86.1	81.6	**83.6**	**0.417 s**

The KMOU corpus having the largest amount of data showed the most reliable results. In the experiments using the KMOU corpus, the performance improved 0.8% when the bi-LSTM model and CRFs were combined; however, the speed declined (approximately 2.7 times). The CNN model was fast; however, its performance was 13.2% worse than bi-LSTM's. The CNN-CRFs model using 64 filters showed a performance of 83%, which was similar to that of bi-LSTM-CRFs. Meanwhile, the CNN-CRFs-2 model using 256 filters showed a performance of 83.6%, which was 0.4% higher, and 27.5% faster than bi-LSTM-CRFs.

In experimental results of CNN-CRFs-2 model using the test dataset of KMOU, the performances of ORG and LOC, which could exhibit ambiguity issues, were low. Furthermore, there were many cases of incorrectly predicting the named entity category in the POH category, which contained many OOV words. In addition, low performances were exhibited by DUR, PNT, and MNY.

H. Y. Yu and Y. J. Ko **[6]** used the KIPSC corpus and demonstrated a performance of F1 85.49%. The study conducted experiments based on morphemes, whereas the proposed model is based on syllables. Furthermore, unlike the proposed model that did not use additional resources, the study utilized additional resources, such as pre-trained word-embedding and named entity dictionary.

M. A. Cheon *et al.* **[10]** used multi-head attention for the ETRI corpus and demonstrated a performance of F1 84.84%. The study used only three categories of personal names, location names, and organization names.

The proposed study demonstrated a performance of F1 83.6% in the experiment with the KMOU corpus. Our method tags NER based on syllables. However, the experimental method was different from that of studies **[6]** and **[10]**. As Korean NER is studied using several kinds of corpus, categories are occasionally different. Furthermore, direct performance comparison is difficult because additional resources such as morpheme-based learning method, pre-learned embedding, and named entity dictionary are used to improve the performance.

5. Conclusion

This paper proposed a syllable-based NER method using CNN, which shows a faster processing speed than the NER using RNN.

In the results of experiments using the ETRI's NER corpus, the CNN model performed 2.4% better than the RNN model. In the case of NER corpus of KIPSC, the performance improved 5.5% using the CNN. However, the overall performance was low because the volume of corpus was small.

In the experiment with KMOU corpus (which had the largest amount of data), the CNN-base model showed a performance of 83.6%, which was 0.4% higher, and 27.5% faster compared to the RNN-based model.

In terms of Korean NER, the models using RNN and the models using CNN showed similar performances. However, the processing speed of proposed model showed approximately 25% improvement compared to that of the model using RNN.

In future studies, a more complex CNN model optimized to Korean NER need to be devised by adding more convolution and pooling layers. In addition, it is necessary to study the effect of Transformer model which is a word embedding model recently proposed, on the speed and accuracy of Korean NER.

Acknowledgment

This work was supported by the National Research Foundation of Korea funded by the Korean government (NRF-2017M3C4A7068187) in 2017. This paper is an extended version of the one entitled "Korean Named Entity Recognizer using CNN-CRFs," which was presented at the 31th Annual Conference on Human and Cognitive Language Technology.

Author Contributions

Conceptualization, Y. S. You and H. R. Park; Methodology, Y. S. You; Software, Y. S. You; Validation, Y. S. You and H. R. Park; Formal Analysis, Y. S. You; Investigation, Y. S. You; Resources, Y. S. You; Data Curation, Y. S. You; Writing—Original Draft Preparation, Y. S. You; Writing—Review & Editing, H. R. Park; Visualization, Y. S. You; Supervision, H. R. Park; Project Administration, H. R. Park; Funding Acquisition, H. R. Park.

References

[1] Y. H. Shin and S. G. Lee, "Bidirectional LSTM-RNNs-CRF for named entity recognition in Korean," Proceedings of the 28th Annual Conference on Human and Cognitive Language Technology, pp. 340-341, 2016 (in Korean).

[2] D. Y. Lee and H. S. Lim, "Korean entity recognition system using Bi-directional LSTM-CNN-CRF," Proceedings of the 29th Annual Conference on Human and Cognitive Language Technology, pp. 327-329, 2017 (in Korean).

[3] H. M. Cho, J. G. Kim, H. S. Kwon, and J. H. Lee, "Named entity recognition using recurrent neural network and convolutional neural network," Proceedings of the KIISE Korea Computer Congress 2017, pp. 636-638, 2017 (in Korean).

[4] Y. J. Jang, T. H. Min, and J. S. Lee, "Named-entity recognition using stacked Bi-LSTM-CRF ensemble model," Proceedings of the KIISE Korea Computer Congress 2018, pp. 2049-2051, 2018 (in Korean).

[5] S. J. Kwon, Y. S. Heo, K. C. Lee, J. S. Lim, H. J. Choi, and J. Y. Seo, "A Korean named entity recognizer using weighted voting based ensemble technique," Proceedings of the 28th Annual Conference on Human and Cognitive Language Technology, pp. 333-335, 2016 (in Korean).

[6] H. Y. Yu and Y. J. Ko, “Expansion of word representation for named entity recognition based on bidirectional LSTM CRFs,” Journal of KIISE, vol. 44, no. 3, pp. 306-313, 2017 (in Korean).

[7] C. Y. Song, S. M. Yang, and S. W. Kang, “Named-entity recognizer based on bidirectional LSTM CRFs using improved word embedding models and dictionaries,” Proceedings of the KIISE Korea Computer Congress 2017, pp. 699-701, 2017 (in Korean).

[8] S. H. Na and J. W. Min, “Character-based LSTM CRFs for named entity recognition,” Proceedings of the KIISE Korea Computer Congress 2016, pp. 729-731, 2016 (in Korean).

[9] J. W. Min and S. H. Na, “Lexicon feature infused character-based LSTM CRFs for Korean named entity recognition,” Proceedings of the 28th Annual Conference on Human and Cognitive Language Technology, pp. 99-101, 2016 (in Korean).

[10] M. A. Cheon, C. H. Kim, H, M. Park, and J. H. Kim, “Character-aware neural networks with multi-head attention mechanism for multilingual named entity recognition,” Proceedings of the 30th Annual Conference on Human and Cognitive Language Technology, pp. 167-170, 2018 (in Korean).

[11] M. A. Cheon, C. H. Kim, H. M. Park, and J. H. Kim, “Evaluating and applying deep learning-based multilingual named entity recognition,” Journal of the Korean Society of Marine Engineering, vol. 42, no. 2, pp. 106-113, 2018.

[12] C. K. Lee, J. S. Kim, J. H. Kim, and H. K. Kim, “Named entity recognition using deep learning,” Proceedings of the 41st KIISE Winter Conference, pp. 423-425, 2014 (in Korean).

[13] C. K. Lee, “Named entity recognition using long short-term memory based recurrent neural network,” Proceedings of the KIISE Korea Computer Congress 2015, pp. 645-647, 2015 (in Korean).

[14] K. H. Choi, H. S. Hwang, and C. K. Lee, “Bio-NER using LSTM-CRF,” Proceedings of the 27th Annual Conference on Human and Cognitive Language Technology, pp. 85-89, 2015 (in Korean).

[15] Y. J. Jang, T. H. Min, and J. S. Lee, “Named-entity recognition using stacked Bi-LSTM-CRF ensemble model,” Proceedings of the KIISE Korea Computer Congress 2018 (KCC 2018), pp. 2049-2051, 2018 (in Korean).

[16] National Institute of Korean Language Named Entity Recognition Corpus, https://ithub.korean.go.kr, Accessed December 12, 2019.

[17] Korea Maritime and Ocean University Named Entity Recognition Corpus, https://github.com/kmounlp/NER, Accessed December 12, 2019.