IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 14, 2020, accepted November 30, 2020, date of publication December 8, 2020,
date of current version December 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043261

Named-Entity Recognition Using Automatic Construction of Training Data From Social Media Messaging Apps

SEUNGWOOK LEE[1] AND YOUNGJOONG KO[2]
[1]Department of Computer Engineering, Dong-A University, Busan 49315 , South Korea
[2]Department of Computer Science and Engineering, Sungkyunkwan University, Suwon 16419, South Korea

Corresponding author: Youngjoong Ko (youngjoong.ko@gmail.com; yjko@skku.edu)

This work was supported in part by the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (A Neural-Symbolic Model for Knowledge Acquisition and Inference Techniques) under Grant 2020-0-00368 and in part by the National Research Foundation of Korea (NRF) grant funded by the Korean Government (MSIT) under Grant 2020R1A2C2100362.

ABSTRACT In recent years, social media messaging app data has served as a precious resource to extract useful information, such as critical clues and evidence in legal trials and criminal investigations. Although these data can be of various types, they are mostly in the form of natural language text. Therefore, to extract information from them efficiently, it is essential to research practical natural language processing approaches. This study proposes applying a deep-learning-based named-entity recognition (NER) system as a natural language processing approach for information extraction to these messaging data. In addition, a system for automatically constructing NER training data is presented using the distant supervision method for the training data of deep-learning models. Because social media messaging app data generally include a significant amount of noise, such as typographical and word-spacing errors, a NER system with robustness against these types of noisy data is required to extract information from the messaging data effectively. The results demonstrate that the proposed approach outperforms that of a NER system with manually labeled training data.

INDEX TERMS Social messaging app data, named-entity recognition, distant supervision, deep learning.

I. INTRODUCTION

With the recent popularization of smartphones and social network service (SNS) applications, private interpersonal communication has become easier through social media messaging (SMM). SMM app data has provided important clues and evidence in legal trials and criminal investigations. Thus, it is crucial to study the extraction of information from these chat data [1].

The SMM app data are of various types, but the most important information is generally in natural language text. Identifying and classifying entity names from proper nouns that have a specific meaning in a given sentence or document is performed by a natural language processing (NLP)-based information extraction approach called named-entity recognition (NER) [2], [3]. In this work, NER is applied to data from SMM apps, and three technical studies are performed to analyze and extract necessary information from the SMM app data.

The first is the application of the distant supervision method. With the rapid development of deep learning and machine learning, recent NER studies have achieved good results. And NER is actively being used, along with other NLP approaches, in various fields. However, its performance varies significantly depending on the training data used for supervised deep-learning and machine-learning techniques. Owing to this problem, training data should be manually constructed with human intervention. However, this approach results in wasted resources, particularly in terms of time and cost. Thus, to solve this issue, the distant supervision method, which is a semi-supervised learning method, was used in this study to construct training data automatically, resulting in automatically labeled data for deep-learning-based NER using the SMM app data. In particular, because SMM apps have various domains for discussion, the proposed distant-supervision-based method

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna D'Ulizia.

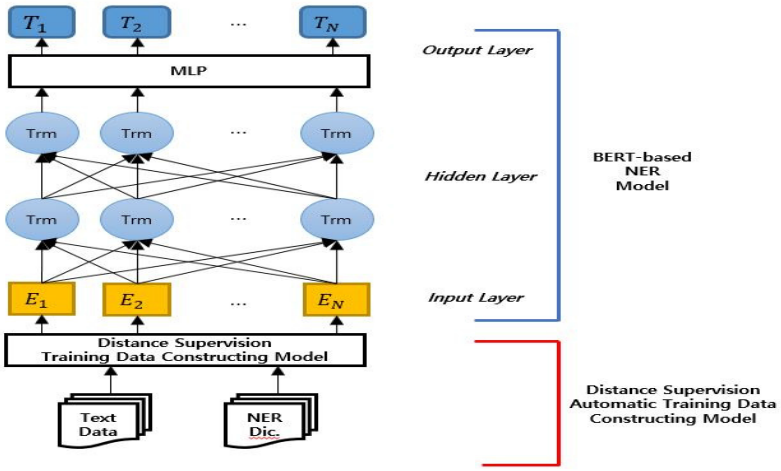

FIGURE 1. Overview architecture of the BERT based NER model using distance supervision.

is more useful for extracting named entities from the data.

The second technical study is the application of syllable-level learning. NER for SMM app data has previously suffered from out-of-vocabulary problems due to many typographical and spacing errors. The NER system proposed herein performs syllable-level learning in deep-learning models, such as bidirectional long short-term memory with a conditional random field (BiLSTM-CRF) [2]. When implementing the BiLSTM-CRF-based NER model with syllable-level learning and some extended features, BiLSTM-CRF-based NER's performance can be improved.

The third technical study is a post-training method using large-sized automatically labeled data before fine-tuning [4], [5]. Post-training in previous studies trained models on domain-specific task data with the same pre-training objectives, such as masked language model and next sentence prediction like BERT, but our post-training method used the same objective for NER tasks on large-sized automatically labeled data. Since BERT (bidirectional encoding representations from transformers) is currently the state-of-the-art model for NER [3], we study how to improve BERT-based NER using automatically labeled data from our distance supervision method (See Figure 1). Consequently, we achieved improved performance when the post-training method with automatically labeled data was applied, and then fine-tuning was conducted on BERT-based NER.

In our experiments, two different-sized datasets, consisting of 4,200 (small) and 63,000 (large) messages, were used as unlabeled data. First, the BiLSTM-CRF-based NER system's performance scores improved by 2.56%p for the small automatically labeled data, 0.82%p for the large automatically labeled data, and 0.56%p for the manually labeled data when syllable-based learning was applied, and the features were extended. In addition, the BiLSTM-CRF-based NER system trained by large automatically labeled data showed 14.14%p improvement compared to the small sample. This result proves that significant improvement can be achieved by applying the distance supervision technique and using large-sized unlabeled data. In particular, the performance of the large auto-labeled data was very close (1.06%p) to the performance of the BiLSTM-CRF-based NER system trained by manually labeled data. However, the BERT-based NER system showed a significant performance difference (7.57%p) between systems trained by large automatically labeled data and manually labeled data. Therefore, we applied post-training to the BERT-based NER system for effectively using both manually and automatically labeled data; the BERT-based NER system with post-training outperformed that trained with the manually labeled data.

The remainder of this article is organized as follows. In Section II, related works are introduced. In Section III, the proposed method is discussed. Section IV describes the data used in the experiments, the experimental environment, and the experimental results. Finally, Section V presents the conclusions.

II. RELATED WORK

A. AUTOMATIC CONSTRUCTION OF TRAINING DATA USING DISTANT SUPERVISION

In the past, studies on distant supervision were conducted to reduce the data construction cost and time. These studies were commonly based on semi-supervised learning methods, which require minimal human intervention for relation extraction, instead of supervised learning methods [6], [7].

In addition to relation extraction, distant supervision studies have also been applied to data constructions, such as NER and emotion analysis, recently. To construct NER's training data automatically, Jingbo *et al.* applied named-entity dictionary refinement and developed a deep-learning-based NER by mapping the result [8]. In [9], the initial NER training data were automatically constructed through dictionary mapping of named entities and raw data from Wikipedia. The study reduced the number of errors in the initial NER training data

by applying active learning with a bagging method and a conditional random field (CRF) method; thus, the NER system's performance was improved. A named-entity dictionary was constructed using the characteristics shown on Wikipedia, and the NER training data was automatically built by mapping the constructed dictionary and raw data. Additionally, the study used Freebased, which is often used in relation extraction, to construct the NER training data automatically; then, the CRF method was applied to the data [10], [11].

Plank *et al.* showed that a particularly good source of not-so-distant supervision is linked websites. In particular, with this supervision source, their method improved on the state-of-the-art for Twitter NER [12]. Fries *et al.* presented SwellShark, a framework for building biomedical NER systems quickly and without hand-labeled data. Their approach observes biomedical resources similar to lexicons as function primitives to construct a generative model for training high-accuracy NER taggers [13]. Wang *et al.* created the CORD-NER dataset with comprehensive NER on the COVID-19 Open Research Dataset Challenge corpus. The CORD-NER dataset covers 75 fine-grained entity types. The CORD-NER annotation is a combination of four sources with different NER methods [14].

B. NAMED ENTITY RECOGNITION (NER)

Research on NER began with the hidden Markov model in the machine-learning study and has developed to use the CRF model [15]–[17]. Further, recent developments have enabled the use of deep learning in NER. Deep-learning-based NER research has been actively conducted as word-embedding techniques, and various deep-learning models have been developed [18]–[20]. Unlike statistical machine learning for NER, deep-learning-based NER has the advantage of being efficiently and effectively applied through letter embeddings and vectorization of various other features, without requiring manual feature extraction through human intervention. In addition, the deep-learning method can consistently increase its performance with an increase in training data. In this study, the BiLSTM-CRF model, known for its high performance, is used; the performance was improved by various feature extensions [21]–[23]. Recently, BERT has shown state-of-the-art performance in sequence labeling and classification NLP tasks [24].

III. PROPOSED METHOD

A. AUTOMATIC CONSTRUCTION OF TRAINING DATA FOR NER USING THE DISTANT SUPERVISION TECHNIQUE

The distant supervision technique has been developed to reduce the time and cost required for data construction using a semi-supervised learning method, which requires minimal human intervention, instead of a supervised learning method. In recent times, this technique has been applied to various fields in addition to relation extraction. Therefore, in this study, it was applied to automatically construct the NER training data using pre-defined rules and a named-entity dictionary. Five commonly used named-entity tags were applied: PERSON, LOCATION, ORGANIZATION, DATE, and TIME. The PERSON, LOCATION, and ORGANIZATION tags were applied with automatic labeling using both the named-entity dictionary and pre-defined rules, and the DATE and TIME tags were applied with automatic labeling using only the pre-defined rules.

TABLE 1. ETRI & Wikipedia Dictionary Refinement Results.

	NUMBER OF ENTRIES
Original ETRI dictionary	1,548,748
Original WIKI dictionary	28,979
Refined ETRI & WIKI dictionary	644,934

1) NAMED-ENTITY DICTIONARY

The named-entity (NE) dictionary used in this study was obtained by refining the NE dictionary established by the Electronics and Telecommunications Research Institute of Korea (ETRI) for the development of answer-type guidelines for the question answering system and generated from Wikipedia by [25]. The ETRI dictionary was analyzed before refinement; it was found that several common nouns that are not commonly used as named entities were included in the dictionary. Further, there was an excess number of names of people with the PERSON tag. For this study, important names were selected by mapping the names in the dictionary to those extracted from Wikipedia. Finally, almost half of the entities that could cause errors, such as names and common nouns, were removed, as seen in Table 1.

2) AUTOMATIC CONSTRUCTION OF NER TRAINING DATA FOR SNS USING A NAMED-ENTITY DICTIONARY AND PRE-DEFINED RULES

The PERSON, LOCATION, and ORGANIZATION tags were automatically labeled by the longest match of strings using the constructed named-entity dictionary and the SMM app data. For the DATE and TIME entity tags, the rules were defined to detect the DATE and TIME entities; they were normalized for effective data analysis. If a word sequence matched a defined rule, it was automatically labeled with the relevant tag. In addition, we used a clue-word dictionary in addition to the named-entity dictionary; in particular, in the SNS environment, honorifics, such as "dear-님 (num)" and "sir-짱 (jjang)," and occupation names were extracted as clue words for the dictionary. For example, once the clue-word dictionary and the SMM app data were matched, the PERSON tag was labeled on the clue word or the preceding word.

Because LOCATION and ORGANIZATION are traditionally highly ambiguous in NER, we attempted to resolve the ambiguity problem partially using the clue dictionary and some linguistic rules in our distance supervision process. First, we extracted sub-words that have ambiguity

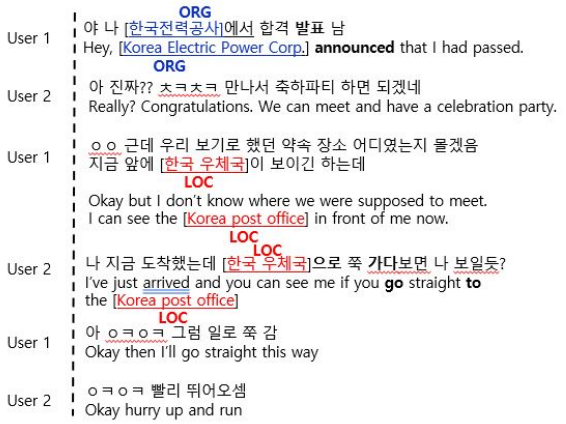

FIGURE 2. Example messages for NER in SMM app data.

but are one-side biased for the tags of entities between LOCATION and ORGANIZATION from the ETRI & WIKI dictionary by hand; LOC/ORG denotes them for an entity with LOCATION biased ambiguity and ORG/LOC for an entity with ORGANIZATION biased ambiguity, such as "우체국 (post office)" for LOC/ORG and "공사 (public corporation)" for ORG/LOC. The numbers of entities for LOC/ORG and ORG/LOC were 133 and 72, respectively. Second, we extracted rules to resolve the ambiguities from labeled training data. We first collected sentences including dictionary entities with extracted sub-words and then investigated them using "동사 (verb)" and "조사 (postposition)" information to figure out which verb and postposition resolved the ambiguity of the dictionary entities between LOCATION and ORGANIZATION. For example, if " 'entity with LOC/ORG or ORG/LOC' + '으로 (to, postposition)' + 가다 (go, verb)"' occurs in a sentence, the entity is labeled as LOC as shown in Figure 2. Finally, the tags of entities that were not determined by the aforementioned rule application were assigned as the biased tag; for example, LOC is for LOC/ORG and ORG is for ORG/LOC.

B. CNN-BiLSTM-CRF-BASED NER SYSTEM

1) BILSTM-CRF NER SYSTEM

The BiLSTM-CRF model structure, which is based on LSTM and CRF, is shown in Figure 3.

LSTM is a particular type of recurrent neural network that is designed with the capability of learning long-term dependencies [26]. An LSTM unit consists of several gates that control the proportion of information both to be given to the memory cell and to be forgotten from the previous state. The following equations were implemented for the LSTM memory cell:

$$i_t = sigmoid(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (1)$$
$$\tilde{c}_t = tanh(W_{xc}x_t + W_{xc}h_{t-1} + b_c) \quad (2)$$
$$c_t = (1 - i_t) \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (3)$$
$$o_t = sigmoid(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (4)$$
$$h_t = o_t \odot tanh(c_t), \quad (5)$$

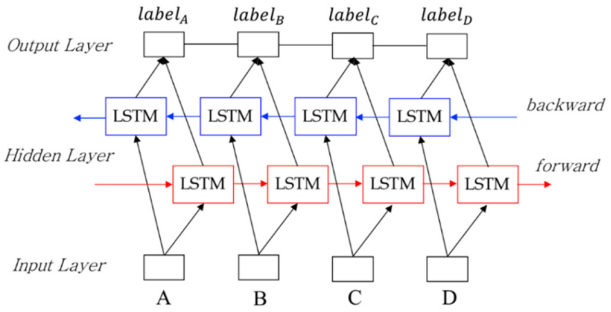

FIGURE 3. BiLSTM-CRF.

where $\odot$ is the element-wise product, and i, o, and c denote the input gate, output gate, and cell vector, respectively. The subscripts of weight matrices, such as W_{hi} and W_{co}, represent the gate in which they are used and the input vector by which they are multiplied. For example, W_{hi} is the matrix for the hidden input vector in the input gate and W_{co} is the matrix for the cell input vector in the output gate. For a given sentence $(x_1, x_2, \cdots, x_n)$ containing n words, each word is represented as a vector, w_t. As an LSTM only models the information flow in one direction, it is generally useful to employ an additional LSTM in the reverse direction. The former is referred to as a forward LSTM and the latter as a backward LSTM; this pair is referred to as BiLSTM [16]. The final hidden state of the BiLSTM model is obtained by concatenating its forward and backward context representations, $h_t = [\vec{h}_t; \overleftarrow{h}_t]$.

Besides, CRF enables LSTM to use neighbor-label information for current-label prediction. For sequence labeling tasks such as NER, it is useful to consider the correlations between labels in neighborhoods and decode the optimal chain of labels for a given input sequence jointly. For example, in NER, I-ORG cannot follow B-PER. Therefore, a label sequence can be jointly decoded using a CRF [27] instead of decoding each label independently.

2) WORD EMBEDDING GENERATION USING A CONVOLUTIONAL NEURAL NETWORK (CNN) AND SYLLABLE EMBEDDING

Recently, there has been diverse research on NER systems, applying various features to the BiLSTM CRF model. This section discusses the effort to generate word embeddings using syllable embedding in the proposed system as a useful feature for SNS datasets. The CNN model, which is primarily used for image processing using deep learning, and syllable embeddings are used to generate word embeddings superior to those of the BiLSTM model in terms of speed.

A current input word is entered into a CNN model as a syllable unit, and the output layer of the CNN model is the syllable-based word representation of the input word, as shown in Figure 4. Each syllable embedding is assigned a vector through random embedding, and a dense vector carrying more information is generated through filtration with the training weight of the convolutional layer. Finally, in the max-pooling layer, final features are created by extracting the

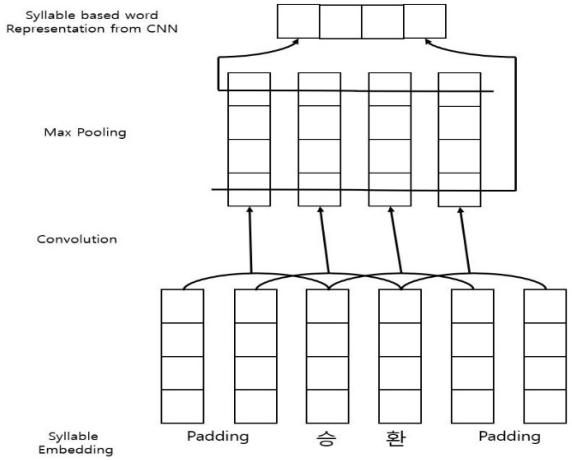

FIGURE 4. Word embedding generation using CNN and syllable embeddings.

TABLE 2. One-Hot Encoding of POS "NNG".

NNG	NNP	NNB	VV	…	JKS	JKO
1	0	0	0	…	0	0

largest value from each syllable vector that passes through the convolutional layer.

3) PART-OF-SPEECH (POS) TAG FEATURES

As the POS sequence is essential in recognizing named entities, the feature vectors that express it are considered important. The one-hot encoding method is generally used to vectorize a POS. For example, if the current POS is NNG, the representation can be shown as in Table 2.

As shown in Table 2, the one-hot encoded POS feature vector's dimension equals the POS number, and the morphology of the current input word is mapped as "1" whereas the rest uses the vector assigned as "0" for the NER features. Because this one-hot encoding for Korean POS information is straightforward and the number of Korean POS tags is commonly more than in English or other languages, we think that the encoding method can be easily applied to other POS structures and languages.

4) EXTENDED WORD REPRESENTATION

There are two steps for obtaining an extended final word representation for NER. First, the concatenation of the word embedding vector from the look-up dictionary and a POS feature one-hot vector is used. Second, a new word embedding is created using syllable-based learning, which takes syllables as inputs to overcome the out-of-vocabulary problem. Eventually, an extended word representation is generated by concatenating word embedding, syllable-based word embedding, and POS feature one-hot vectors, as shown in Figure 5.

C. BERT BASED NER SYSTEM

We used BERT as our pre-trained language model for a NER task. At the end of 2018, Google's BERT was the best-performing model for the GLUE Benchmark [4], [5]. In contrast to previous language models, they used a deeply bidirectional architecture for their transformer; the model receives the whole sentence (or sentence pair) as input, and each cell depends on the context of the previous and subsequent word in the sequence. The non-sequential input makes the next-word prediction task impossible. Instead, Devlin *et al.* trained the model to predict masked words in the input sentence [4]. They also trained it to identify if two sentences can occur next to each other for further cross-sentence context. In this research, KorBERT[1] was used on the following hyperparameters: L (layer number) = 12, H (dimension of hidden layer) = 768, A (number of multi-heads) = 12. The overall structure of KorBERT is shown in Figure 6.

1) POST-TRAINING USING AUTOMATICALLY LABELED DATA

To use the automatically labeled data from distance supervision for BERT effectively, we constructed post-training with the large-sized automatically labeled data before fine-tuning using the same objective as NER. Although the automatically labeled data are noisy data with some errors, they can contribute to NER's improvement by their usage of post-training before fine-tuning. As a result, the proposed method can achieve 1.31%p higher performance than BERT-based NER with the manually labeled data.

IV. EXPERIMENTS

A. EXPERIMENTAL DATA AND SETTINGS

In this study, data were collected from KakaoTalk, a popular SMM app in South Korea. The collected chatroom data were classified into six categories: IT, study, exercise, games, hobbies, and others. The total number of collected chat messages was 12,844,803, and the processing unit was a message. To use these data for NER, the data were randomly extracted from each category. A total of 70,000 messages were extracted for the training data, and 10% of 70,000 messages (7,000 messages) were randomly selected and manually labeled using the BIO tagging scheme by three trained annotators, which is a commonly used method in sequence labeling tasks. The 7,000 labeled messages were composed of 4,200 training messages and 2,800 test messages; the remaining 63,000 messages were used as unlabeled data for automatic labeling by the proposed distance supervision method, as shown in Table 3. To measure inter-rater reliability for qualitative items, Cohen's kappa value was calculated as 0.81, and it was evaluated as "good agreement." All three annotators discussed any conflict until an agreement was reached.

The evaluation metrics we used to measure the proposed NER system's performance were precision, recall, and F1. We used the following terminology for describing our evaluation metrics. "Correct" is the number of named entities (NEs) that match the correct answer, and the count of "Correct" is

[1] aiopen.etri.re.kr

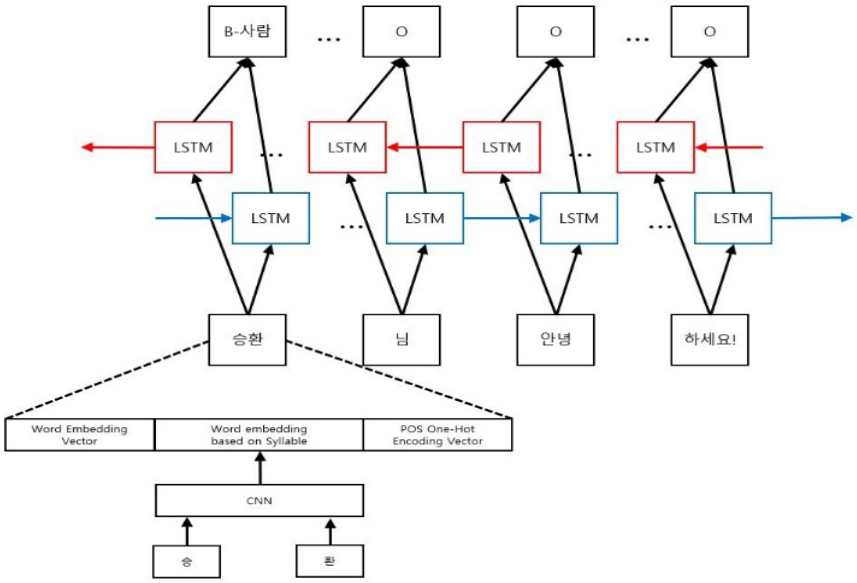

FIGURE 5. CNN-BiLSTM-CRF NER Model.

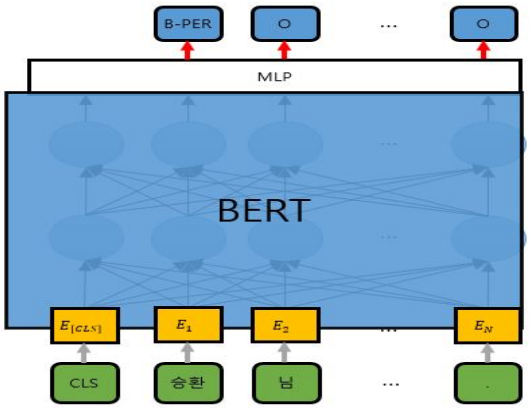

FIGURE 6. BERT-based model for the NER task.

TABLE 3. Distribution of Labeled and Unlabeled Data Used for Training and Testing.

	Training	Testing	Total
Labeled data	4,200	2,800	7,000
Unlabeled data (distance supervision)	63,000		63,000

measured using exact matches of the NEs, not single words. ''Predict'' is the number of NEs predicted by the NER system. ''Answer'' is the number of gold-standard NEs that we have to recognize in test data. Precision measures the quality of predictions, and it is represented as the ratio of the number of predicted NEs that are correct answers to the number of NEs predicted by the proposed NER system. Recall measures how much the proposed model can capture the actual answers in test data. It is calculated as the number of correctly predicted NEs divided by the number of real answers in the test data. F1-score is defined as the harmonic average of precision and recall.

$$\begin{aligned} \text{precision} &= \frac{\text{Number of correct predictions}}{\text{Number of system predictions}} \\ &= \frac{\text{Correct}}{\text{Predict}} \end{aligned} \tag{6}$$

$$\begin{aligned} \text{recall} &= \frac{\text{Number of correct predictions}}{\text{Number of golden standards}} \\ &= \frac{\text{Correct}}{\text{Answer}} \end{aligned} \tag{7}$$

$$\text{F1} - \text{score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{8}$$

The hyperparameters used in this study were as follows. The vector size of the word embedding was set to 64 dimensions, the hidden state of LSTM was 100 dimensions, and the dropout rate was 0.25. The character embedding was set to 30 dimensions, and the number of CNN filters required to generate syllables was set to 30. The filter window size was set to three, and the size of the hidden layer in LSTM for generating syllables was set to 32. The maximum word length for CNN input was set to 100. Lastly, the POS feature vector size was set to 46. The performance evaluation methods used to evaluate the NER module were precision, recall, and F1-score.

B. EXPERIMENTAL RESULTS

1) EFFECTIVENESS OF SYLLABLE EMBEDDING AND ONE-HOT ENCODING VECTOR IN BiLSTM-CRF

Table 4 shows the changes in performance from the baseline model, the selected model with BiLSTM CRFs, to the ''baseline + syllable (CNN) + POS'' model. When syllable embedding with CNN and POS features are added to the

TABLE 4. Experimental Results Using the Small Automatically Labeled Data With Distant Supervision.

Auto (Train 4,200/ Test 2,800)	Precision	Recall	F1
Baseline (BiLSTM CRFs)	60.05%	40.00%	48.12%
Baseline + syllable (CNN) + POS	**61.31%**	**43.19%**	**50.68%**

TABLE 5. Experimental Results Using the Large Automatically Labeled Data With Distant Supervision.

Auto (Train 67,200 /Test 2,800)	Precision	Recall	F1
Baseline (BiLSTM CRFs)	66.58%	62.12%	64.28%
Baseline + syllable (CNN) + POS	**66.27%**	**63.97%**	**65.10%**

baseline, the proposed method's performance improves by 2.56%p with the small automatically labeled data (4,200 messages) generated by distance supervision. In this case, the 4,200 automatically labeled messages were originally from 4,200 manually labeled messages. For this experiment, they were used as unlabeled data by omitting the labels.

Table 5 shows the performance changes from the baseline model to the "baseline + syllable (CNN) + POS" model on the large sample of automatically labeled data, consisting of 67,200 messages (the sum of 63,000 unlabeled messages and the small sample of 4,200 automatically labeled messages shown in Table 4). When syllable embedding with CNN and POS features are added to the baseline, the proposed method improved by 0.82%p on the large automatically labeled data (67,200 messages) generated by distance supervision. In particular, 14.42%p improvement was achieved on the large automatically labeled data compared to the small sample.

Table 6 shows a performance improvement of 0.56%p between the baseline and "baseline + syllable (CNN) + POS" models in the manually labeled data. In all the three individual datasets, for both the small and large automatically labeled and manually labeled data, the proposed methods show similar improvement patterns, as shown in Tables 4–6.

2) EFFECTIVENESS OF BERT ON SMALL AND LARGE AUTOMATICALLY LABELED AND MANUALLY LABELED DATA

We experimented with the same experimental settings for BERT using small and large automatically labeled and manually labeled data. We obtained similar results, as shown in Table 7 and Figure 7. BERT showed a 7.57%p difference in performance compared to NER with manually labeled data. Note that BiLSTM-CRFs showed similar performance (a difference of 1.06%p) to NER with manually labeled data, as shown in Tables 5 and 6.

TABLE 6. Experimental Results on Manually Labeled Data.

Manual (Train 4,200 / Test 2,800)	Precision	Recall	F1
Baseline (BiLSTM CRFS)	78.03%	56.58%	65.60%
Baseline + syllable (CNN) + POS	**73.31%**	**60.28%**	**66.16%**

TABLE 7. Experimental Results of BERT on Small and Large Automatically and Manually Labeled Data.

Data	Precision	Recall	F1
Small Auto	67.97%	47.92%	**56.21%**
Large Auto	58.82%	59.23%	**61.52%**
Manual	79.17%	61.29%	**69.09%**

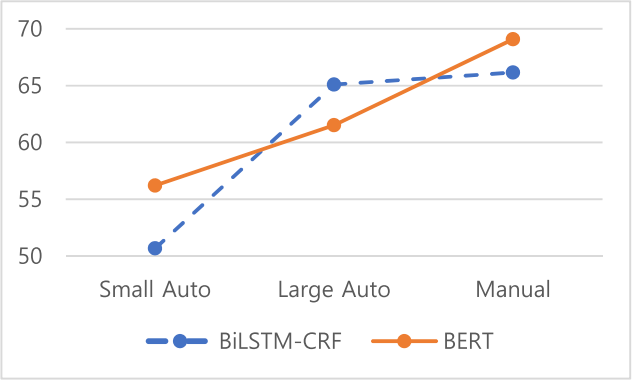

FIGURE 7. Experimental results from automatically and manually labeled data.

TABLE 8. Comparison of F1 Scores of BERT-Based NER Systems on Simple Combination and Post-Training.

	Simple combination	Post-training	Manual data
F1 score	58.99%	***70.4.%***	69.09%

3) HOW TO USE AUTOMATICALLY LABELED DATA TO IMPROVE THE BERT-BASED NER SYSTEM

In the previous section, BERT-based NER did not achieve sufficient performance on the large automatically labeled data when compared to the manually labeled data. Thus, we attempted to improve our BERT-based NER system using both the automatically and manually labeled data. Post-training was applied to our BERT-based NER system,

TABLE 9. Abbreviation Table.

Abbreviation	Full name
BERT	bidirectional encoding representations from transformers
BiLSTM-CRF	bidirectional long short-term memory with a conditional random field
CNN	convolutional neural network
CRF	conditional random field
ETRI	electronics and telecommunications research institute of Korea
HMM	hidden markov model
LSTM	long short term memory
NER	named entity recognition
NLP	natural language processing
OOV	out-of-vocabulary
POS	part of speech
RNN	recurrent neural network
SMM	social media messaging
SNS	social network service

and we finally obtained better performance (by 1.31%p) than BERT-based NER with only the manually labeled data. In particular, even though the BERT-based NER system with a simple combination of automatically and manually labeled data showed low performance, the BERT-based NER system using post-training achieved much better performance (a 11.41%p improvement) than the system with a simple combination, as shown in Table 8.

V. CONCLUSION

In this study, rather than using the news data and Wikipedia document data used in prior NER studies, our deep-learning-based NER model was applied to SMM app data, which include a large amount of natural language text, new words, and unfixable errors. As a result, when using syllable embedding for CNN and POS features, superior performance results were obtained compared to the baseline BiLSTM CRFs-based NER model. Besides, automatic training data were constructed using the distant supervision method, which uses adaptive rules and a named-entity dictionary on the SMM app data. Performance similar to the manually labeled data was achieved in the BiLSTM CRFs-based NER system even for the data mixed with various domain information, such as the SMM app data used in this study; the BERT-based NER system using automatically labeled data outperformed that trained by the manually labeled data when post-training was applied to it.

In the future, in addition to examining the characteristics of the SMM app data, a study on automatically constructing more precise training data will be conducted using more refined rules and a more extensive dictionary.

APPENDIX

See table 9.

REFERENCES

[1] D. Kong, C. Fu, J. Yang, D. Xu, and L. Han, "The impact of the collective influence of search engines on social networks," *IEEE Access*, vol. 5, pp. 24920–24930, 2017.
[2] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*. [Online]. Available: https://arxiv.org/abs/1508.01991
[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805
[4] H. Xu, B. Liu, L. Shu, and P. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis," in *Proc. NAACL-HLT*, 2019, pp. 2324–2335.
[5] T. Whang, D. Lee, C. Lee, K. Yang, D. Oh, and H. Lim, "An effective domain adaptive post-training method for BERT in response selection," in *Proc. Interspeech*, Oct. 2020, pp. 1–5.
[6] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *Proc. 4th Int. Joint Conf. Natural Lang. Process.*, 2009, pp. 1003–1011.
[7] M. Surdeanu, D. McClosky, U. Tibshirani, J. Bauer, A. X. Chang, V. I. Spitkovsky, and C. D. Manning, "A simple distant supervision approach for the TAC-KBP slot filling task," in *Proc. Text Anal. Conf. Workshop (TAC)*, Gaithersburg, MD, USA, 2010, pp. 1–5.
[8] J. Shang, L. Liu, X. Ren, X. Gu, T. Ren, and J. Han, "Learning named entity tagger using domain-specific dictionary," 2018, *arXiv:1809.03599*. [Online]. Available: http://arxiv.org/abs/1809.03599
[9] S. Lee, Y. Song, M. Choi, and H. Kim, "Bagging-based active learning model for named entity recognition with distant supervision," in *Proc. Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2016, pp. 321–324.
[10] S. Yan, W. S. Spangler, and Y. Chen, "Chemical name extraction based on automatic training data generation and rich feature set," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 10, no. 5, pp. 1218–1233, Sep. 2013.
[11] J. Kim, Y. Ko, and J. Seo, "A bootstrapping approach with CRF and deep learning models for improving the biomedical named entity recognition in multi-domains," *IEEE Access*, vol. 7, pp. 70308–70318, 2019.
[12] B. Plank, D. Hov, R. McDonald, and A. Sogaard, "Adapting taggers to Twitter with not-so-distant supervision," in *Proc. COLING*, 2014, pp. 1783–1792.
[13] J. Fries, S. Wu, A. Ratner, and C. Ré, "SwellShark: A generative model for biomedical named entity recognition without labeled data," 2017, *arXiv:1704.06360*. [Online]. Available: http://arxiv.org/abs/1704.06360
[14] X. Wang, X. Song, B. Li, Y. Guan, and J. Han, "Comprehensive named entity recognition on CORD-19 with distant or weak supervision," 2020, *arXiv:2003.12218*. [Online]. Available: http://arxiv.org/abs/2003.12218
[15] P. Sharma, U. Sharma, and J. Kalita, "Named entity recognition in assamese using CRFS and rules," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Oct. 2014, pp. 15–18.
[16] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML*, 2001, pp. 282–289.
[17] Y. Jin, J. Xie, W. Guo, C. Luo, D. Wu, and R. Wang, "LSTM-CRF neural network with gated self attention for chinese NER," *IEEE Access*, vol. 7, pp. 136694–136703, 2019.
[18] N. Zhang, F. Li, G. Xu, W. Zhang, and H. Yu, "Chinese NER using dynamic meta-embeddings," *IEEE Access*, vol. 7, pp. 64450–64459, 2019.
[19] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," 2015, *arXiv:1511.08308*. [Online]. Available: http://arxiv.org/abs/1511.08308
[20] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," 2016, *arXiv:1603.01360*. [Online]. Available: http://arxiv.org/abs/1603.01360
[21] L. Zhao, X. Qiu, Q. Zhang, and X. Huang, "Sequence labeling with deep gated dual path CNN," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 12, pp. 2326–2335, Dec. 2019.
[22] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF," 2016, *arXiv:1603.01354*. [Online]. Available: http://arxiv.org/abs/1603.01354
[23] H. Yu and Y. Ko, "Expansion of word representation for named entity recognition based on bidirectional LSTM CRFs," *J. KIISE*, vol. 44, no. 3, pp. 306–313, Mar. 2017.

[24] Z. Wang, P. Ng, X. Ma, R. Nallapati, and B. Xiang, "Multi-passage BERT: A globally normalized bert model for open-domain question answering," 2018, *arXiv:1908.08167*. [Online]. Available: https://arxiv.org/abs/1908.08167
[25] S. Bae and Y. Ko, "Automatic construction of korean named entity dictionaries from Wikipedia," *J. Korean Inst. Inf. Sci. Eng.*, vol. 16, no. 4, pp. 492–496, Apr. 2010.
[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[27] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," *Neural Netw.*, vol. 18, no. 5, pp. 602–610, 2005.

SEUNGWOOK LEE received the M.S. degree from the Department of Computer Engineering, Dong-A University, Busan, South Korea, in 2020. His research interests include natural language processing, information retrieval, machine learning (deep neural network), named entity recognition, and big data analysis.

YOUNGJOONG KO received the Ph.D. degree from the Department of Computer Science, Sogang University, Seoul, South Korea, in 2003. From 1996 to 1997, he was with LG-EDS. Since 2004, he joined the Faculty of Dong-A University, Busan, where he leaded the Intelligent System Laboratory, Department of Computer Engineering. He was also with the Computational Linguistics and Information Processing Laboratory (CLIP), University of Maryland at College Park, College Park, as a Visiting Scholar from 2011 to 2012. In 2019, he moved Sungkyunkwan University, Suwon. His research interests include natural language processing, machine learning (deep neural network), spoken dialogue systems, information retrieval, and big data analysis.