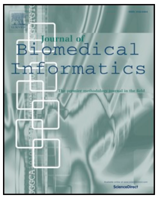
Journal of Biomedical Informatics 103 (2020) 103381

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin



Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition

Minsoo Cho[1,2], Jihwan Ha[2], Chihyun Park, Sanghyun Park*

Yonsei University, Department of Computer Science, Republic of Korea

ARTICLE INFO

Keywords:
Biomedical named entity recognition
Information retrieval
Feature embedding
Attention mechanism

ABSTRACT

With the rapid advancement of technology and the necessity of processing large amounts of data, biomedical Named Entity Recognition (NER) has become an essential technique for information extraction in the biomedical field. NER, which is a sequence-labeling task, has been performed using various traditional techniques including dictionary-, rule-, machine learning-, and deep learning-based methods. However, as existing biomedical NER models are insufficient to handle new and unseen entity types from the growing biomedical data, the development of more effective and accurate biomedical NER models is being widely researched. Among biomedical NER models utilizing deep learning approaches, there have been only a few studies involving the design of high-level features in the embedding layer. In this regard, herein, we propose a deep learning NER model that effectively represents biomedical word tokens through the design of a combinatorial feature embedding. The proposed model is based on Bidirectional Long Short-Term Memory (bi-LSTM) with Conditional Random Field (CRF) and enhanced by integrating two different character-level representations extracted from a Convolutional Neural Network (CNN) and bi-LSTM. Additionally, an attention mechanism is applied to the model to focus on the relevant tokens in the sentence, which alleviates the long-term dependency problem of the LSTM model and allows effective recognition of entities. The proposed model was evaluated on two benchmark datasets, the JNLPBA and NCBI-Disease, and a comparative analysis with the existing models is performed. The proposed model achieved a relatively higher performance with an F1-score of 86.93% in case of NCBI-Disease, and a competitive performance for the JNLPBA with an F1-score of 75.31%.

1. Introduction

With the rapid development of technology in the biomedical field, the amount of biomedical data has grown exponentially. Regarding the efficient extraction of useful information from such data, a great deal of progress has been made in biomedical text mining and Natural Language Processing (NLP) over the past decades. Representative text mining research in the clinical field include engines for automatic identification, analysis, and extraction of specific types of clinical information from medical reports or biomedical literature [2,8]. Named Entity Recognition (NER) is another text mining technique, which involves identifying named entities from a sequence of words and classifying them into predefined categories. It is not only applicable in various NLP tasks such as relation extraction [32], document classification [16], and question and answering systems [26], but also in biomedical literature to identify biological entities (e.g., chemicals, diseases, genes, and proteins). The extracted information is employed in various types of research, including gene discovery, drug development, and disease treatment. The demand for automatic biomedical NER (bioNER) system will continue to grow as it is widely applied in knowledge discovery and data mining analysis

Several studies [3,18] have shown that the NER of a general domain achieves a high F1-score, whereas the performance of bioNER is comparatively low owing to the following factors. First, there are several entity names with variant spelling forms. For instance, the following three words refer to the same chemical: “N-acetylcysteine,” “N-acetyl-cysteine,” and “NacetylCysteine” [1]. Second, biomedical entities have inconsistent use of prefixes and suffixes consisting of letters, numbers, special characters, and Greek letters (e.g.,“Dbf4p”, “Cbp/p300-interacting transactivator”) [1]. Finally, there are cases in which different entities use the same acronym (e.g., “angiotensin converting enzyme,” “affinity capillary electrophoresis,” and “acetylcholinesterase,” are

* Corresponding author.

E-mail addresses: minsoo0104@yonsei.ac.kr (M. Cho), jihwanha@yonsei.ac.kr (J. Ha), chihyun.park@yonsei.ac.kr (C. Park), sanghyun@yonsei.ac.kr (S. Park).

[1] First Author.

[2] These authors contributed equally to this work.

https://doi.org/10.1016/j.jbi.2020.103381
Received 27 February 2019; Received in revised form 26 January 2020; Accepted 27 January 2020
Available online 28 January 2020

denoted as, "ACE"), which generates ambiguity among entities [20].

There are various methods for addressing the aforementioned difficulties in NER, including dictionary- [33], rule- [32], machine learning- [27,30], and deep learning-based methods [10]. In dictionary-based methods, terms in the dictionaries are simply matched with the words in the target sequence for entity extraction. Although this method is simple, the consistent increase in the number of biomedical entities and the variety of notations make entity extraction difficult. In rule-based methods, entity extraction tends to show high performance when applied to only one specific domain. For machine learning-based methods, the model efficiently performs entity extraction using various algorithms and statistical models [12]. However, both rule- and machine learning-based methods are highly dependent on feature engineering, which is not only labor-intensive and time consuming, but also requires a substantial amount of domain knowledge in the biomedical field. Unlike previous approaches that require laborious human tasks for crafting features, deep learning methods automatically extract the best representative features using neural networks. Various neural network architectures [5,9,10,34] have achieved state-of-the-art performance on several biomedical datasets.

Among deep learning methods, the convolutional neural network (CNN) is a well-known architecture that is widely used to capture local information within the given words in biomedical contexts. With the application of deep learning models, hybrid models [23] have also been developed to enhance the prediction accuracy by combining a neural network with a Conditional Random Field (CRF) [17] algorithm. Among such hybrid models, bidirectional long short-term memory (bi-LSTM) [4,11] is commonly used to handle sequential data. In the embedding layer of the bi-LSTM model, not only word-level but also character-level embedding vectors are used as input to handle out-of-vocabulary (OOV) words [18], which are unknown words that are absent in the trained vocabulary set. Numerous studies have proposed different types of CNN- and bi-LSTM-based models for extracting meaningful character-level embedding.

Recently, a number of studies have been proposed that use transfer learning [29] and multi-task learning, which leverage additional information by training a single model for multiple tasks or using pre-trained weights of auxiliary tasks. In study [19], a biomedical version of Bidirectional Encoder Representations from Transformers (BERT) architecture [6] is applied to transfer knowledge pre-trained on a massive amount of unlabeled corpus. In other studies on multi-task learning [36,37], datasets of different types of entities are trained on the same model to leverage information obtained from related tasks. These methods have yielded promising results by utilizing information obtained from large amount of data. Similarly, in this research, we utilize word vectors trained from a large biomedical corpus to capture semantic information and the relationship between words. Moreover, we construct a simple and technically strong model through the design of high-level features by combining character features in the embedding layer. Consequently, the model effectively represents the contextual meaning of each token by capturing morphological information from biomedical terms with unfamiliar and complex structures.

The proposed model consists of CNN and bi-LSTM for a character-level representation with a word-level representation to effectively design high-level combinatorial feature embedding. By integrating two different character-level features extracted from the CNN and bi-LSTM, the token representation captures both the local and global features at the character-level of the embedding. The three representations that include word-level, character-level CNN and character-level bi-LSTM are then concatenated and fed into a fully connected network to adaptively learn the mixture of each representation. Additionally, we apply an attention mechanism to the bi-LSTM-CRF model to compute the similarities between the input tokens to focus on related tokens in the sentence for predicting the tag of the current token. We validated our model with two benchmark datasets, JNLPBA [13] and NCBI-Disease [7], and compared its performance with that of previous models.

The main contributions of the proposed model are summarized as follows:

- A technically simple architecture is proposed that utilizes combinatorial feature embedding and attention mechanism which focuses on relevant parts of the current token in a sequence of words to efficiently recognize entities.
- Effective word representation through the design of a high-level feature embedding utilizing character-level CNN and bi-LSTM, which effectively captures the local and global information of a word token.
- Comparison with state-of-the-art methods on two publicly available datasets demonstrates the empirical strength of our work.
- We study the impact of each module (embeddings, attention mechanism) and the effectiveness of their combination.

The remainder of the paper is organized as follows. In Section 2, we explain the architecture of the bi-LSTM-CRF model and the proposed method. In Section 3, the experimental setup is described in detail. In Section 4, we evaluate the performance of the proposed model and compare it with that of previous methods. In the final section, we summarize the proposed model and discuss possible improvements for future work.

2. Model architecture

In this section, before each component of the proposed model is explained, the overall process is briefly described. The overall architecture of the proposed model is illustrated in Fig. 1.

1. A sentence is given as input to the model. Two different character-level word embeddings of each word token in the sentence are then obtained by utilizing the CNN and bi-LSTM model.
2. Word vectors that are pre-trained from biomedical corpora are used as initialization for word-level embeddings.
3. The obtained word-level embedding and two different character-level embeddings are concatenated and fed into a simple fully connected network to form a 200-dimensional vector, which is used as the input in the next step.
4. The sequence of final word embeddings is fed into the bi-LSTM-CRF model with an attention layer to predict the possible tags for each input sentence.

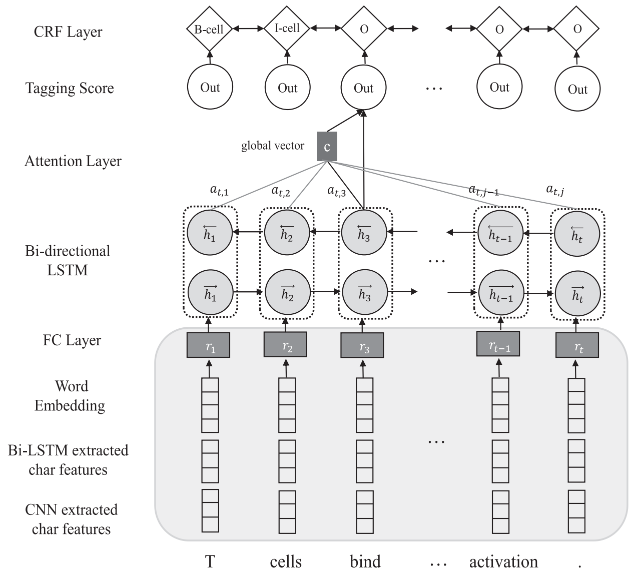


Fig. 1. Overall architecture of the proposed model.

2.1. Word embedding

Word embedding is a distributed word representation that maps words into low-dimensional vectors [25]. The advantage of using a word vector is that it captures the semantics of words or relationships between them. In our model, we utilize publicly available pre-trained word vectors from large biomedical corpora [24]. Because several words in the biomedical literature are not commonly used, the use of pre-trained word vectors trained on biomedical corpora is effective for representing the biological meaning of each entity.

2.2. Character embedding

In addition to word embedding, character-level embedding is used to represent input tokens. It is used for dealing with OOV words that do not exist in the trained word vectors. Especially in the biomedical domain, several unique words exist in irregular forms, and character-level embedding is useful for effectively extracting morphological information of each word token. We use two different neural networks the CNN and bi-LSTM to extract the character-level features.

2.2.1. Character-level CNN model

CNNs, which were initially applied in image processing [15], are now also widely used in several NLP tasks. We apply CNN to effectively extract local information from the characters of each input token. Each character in the word token is mapped to a character vector, as shown in the first step of Fig. 2. The filters with different sizes are then applied to the embedding matrix to capture important features of adjacent inputs. We use three different filter widths in the convolution procedure to capture various features. For the last procedure of CNN, a max-pooling operation is performed to extract a single feature for all feature maps. The output features are then concatenated to represent each word, which preserves the local information.

2.2.2. Character-level Bi-LSTM model

Bidirectional LSTM models are also employed for extracting character-level features. As shown in Fig. 3, we apply the bi-LSTM over the sequence of character embedding for each word and concatenate the two final hidden states from the forward and backward LSTM to obtain a fixed-size vector representing a word token. By using bidirectional hidden states, the model can preserve information from both the past and the future. Additionally, the bi-LSTM model effectively captures the global features of each word token.

2.3. Integrating character and word-level representations

We apply a fully connected network in the embedding layer of the model to integrate character- and word-level representations. First, we concatenate two different character-level representations extracted from CNN and bi-LSTM with word-level representation. The three concatenated representations are then fed into a fully connected network, without an activation function for the JNLBPA dataset and the ReLU function for the NCBI-Disease dataset, which are the results of hyperparameter tuning. Subsequently, the output from the fully connected network is used as the final representation for each input token. By utilizing this method, the model generates a word vector with the most salient features from each type of representation. As indicated by Eq. (1), the three embedding vectors are concatenated and represented as a word vector x_t. The operator ⊕ represents the concatenation of the embedding vectors. As indicated by Eq. (2), the input word vector x_t is fed into the fully connected network with the weight matrix W_i and bias vector b. The word embedding x_t is the output of the fully connected network with a vector size of 200.

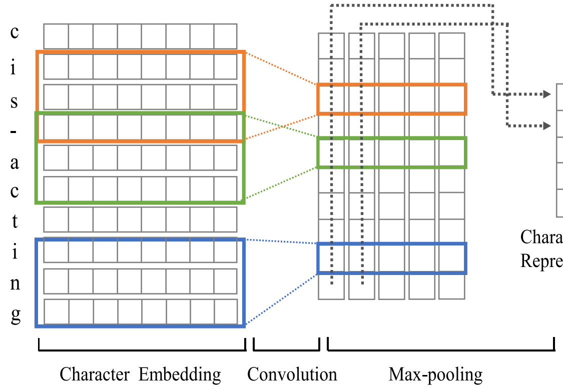


Fig. 2. Character-level CNN model.

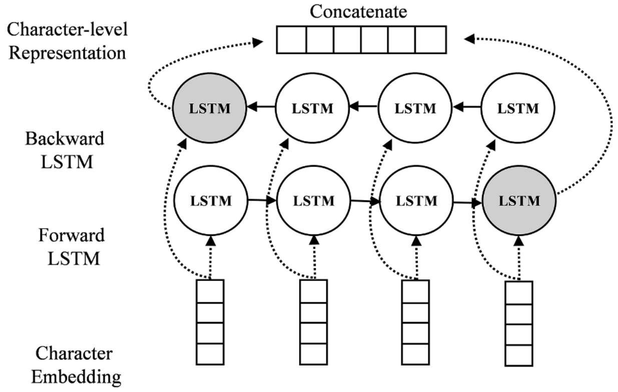


Fig. 3. Character-level bi-LSTM model.

$$x_t = [v_{word} \oplus v_{cnn} \oplus v_{bi-lstm}] \tag{1}$$

$$w_t = f(W_i x_t + b) \tag{2}$$

2.4. Bi-LSTM-CRF model

In the unidirectional LSTM model, only information from the previous words is captured, as tokens are fed into the network from left to right. However, in the bi-LSTM model, by processing the input in the forward and backward directions, the one can keep track of information from both directions. Given the input sentence X = {x_1, x_2, $x_3 \cdots x_t$}, the hidden states $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ of the forward and backward LSTM outputs, respectively, are computed in both directions. They are concatenated as $h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right]$ and used as a word representation of each word token in step t.

We add a CRF on top of the bi-LSTM layer, as shown in Fig. 4. Even without the CRF layer, the bi-LSTM model can independently predict the label of each word. However, with the CRF layer, the dependencies among adjacent tags are considered. The CRF layer jointly decodes the best tag path using the state transition matrix.

$$s([x]_1^T [y]_1^T) = \sum_{t=1}^{T} \left(T_{y_{t-1},y_t} + M([S]_1^T)_{y_t,t} \right) \tag{3}$$

In Eq. (3), T_{y_{t-1},y_t} represents the transition score of the transition from label $y_{t-1,}$ to label y_t. Given the input sentence $x = \{x_1, x_2, x_3 \cdots x_t\}$, the transition score T_{y_{t-1},y_t} and $M([S]_1{}^T)_{y_t,t}$, which is the tagging score of the bi-LSTM network for the t^{th} word with label y_t, are summed to represent the final score of each sentence.

After the calculation of the final score, as indicated by Eq. (4), the softmax function is applied to obtain the conditional probability of t^{th} word of the sequence with label y_t, over all possible paths $Y(x)$, for sentence x.

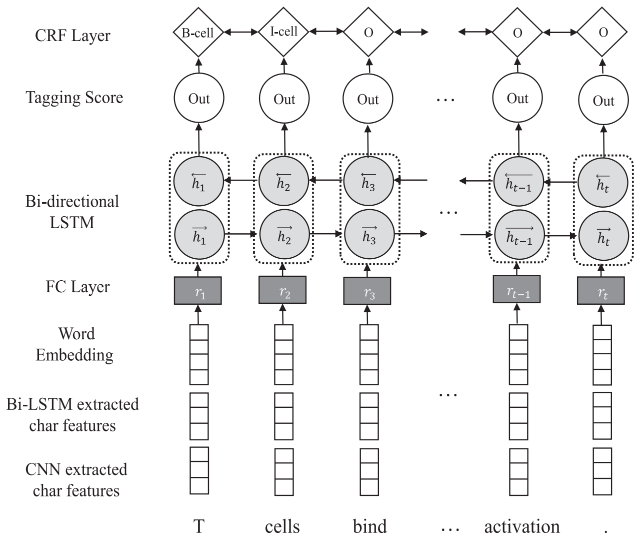


Fig. 4. Bi-LSTM-CRF model.

$$p(y|x) = \frac{\exp(s([x]_1^T,[y]_1^T))}{\sum_{y' \in Y(x)} \exp(s([x]_1^T,[y']_1^T))} \tag{4}$$

During training, the model maximizes the log probability of Eq. (4). After training, the model decodes the best output sequence with the maximum final score as follows:

$$\hat{y} = \underset{y' \in Y(x)}{\text{argmax}} s([x]_1^T,[y']_1^T) \tag{5}$$

2.5. Bi-LSTM-CRF model with attention

Additionally, the model contains an attention layer between the bi-LSTM and CRF layers, which is illustrated in Fig. 1. The attention mechanism, which was first introduced in the area of computer vision, is now widely used in various fields [22,35] including NLP. In NLP, the attention mechanism improves performance by focusing on the relevant parts of a sequence of words more than on the irrelevant parts when predicting. Moreover, when the attention mechanism is applied to the LSTM layer, it alleviates the long-term dependency issue that can occur in the long input sequences.

There is a sequence labeling study using an attention mechanism to combine word-level and character-level representations [28]. In a recent study of chemical NER, an attention mechanism is used to capture similar entities at the document level to solve the problem of tagging inconsistency [21]. Motivated by the previous study [21], in which an attention mechanism was incorporated at the document level, we apply an attention mechanism at the sentence level to enhance the model performance. By applying the attention mechanism, the model is trained to focus on relevant tokens in each sentence, and the similarity information is employed to effectively predict the label of each word. For instance, if in the following given sentence “Octamer-binding proteins from C or HeLa cells stimulate transcription of …,” the tokens “Octamer-binding” and “proteins,” which are tagged as “B-protein” and “I-protein” in the standard corpus, have a high attention score relative to other token pairs, this can increase the likelihood that the model will determine the two tokens as the correct boundary of the protein entity.

In the attention layer, for the input sentence $x = \{x_1, x_2, x_3 \cdots x_t\}$, the score function is calculated for each target word x_t and all other words x_i in the sentence to calculate similarities between two words. We performed various experiments with different score functions, including dot product, Euclidean distance, and Manhattan distance function. Among the score functions, we chose the Manhattan distance function, which exhibits the best performance and involves a simple calculation. In Eq. (6), $W_a$is a trainable weight matrix.

$$score(x_t, x_i) = W_a|x_t - x_i| \tag{6}$$

The softmax function is then used to normalize the score, which generates the attention weight $\alpha_{t,i}$,conditioned on the target words.

$$\alpha_{t,i} = \frac{\exp(score(x_t, x_i))}{\sum_k \exp(score(x_t, x_k))} \tag{7}$$

Subsequently, we generate a context vector $c_t$for each target word by computing the weighted sum of the hidden states multiplied by the attention weight $\alpha_{t,i}$.

$$c_t = \sum_i \alpha_{t,i} h_i \tag{8}$$

The context vector and hidden states from the bi-LSTM model are then concatenated to form a word representation $o_t = [h_t; c_t]$ for each target word. Finally, the word representation o_t is fed into a fully connected network for computing the tagging score. By utilizing the global vector c_t, the model can capture global information of the entire input sequence and accurately predict the label of each word token. The architecture of the bi-LSTM-CRF model with the attention layer is shown in Fig. 1.

3. Experiment

3.1. Datasets

We used two publicly available benchmark datasets, the JNLPBA [13] and NCBI-Disease [70], to evaluate the model in comparison with other competitive models. The JNLPBA corpus consists of 22,402 sentences (18,546 for the training set and 3856 for the test set) from 2400 abstracts in the MEDLINE database. The NCBI-Disease corpus includes 6892 disease mentions (5145 in the training set, 787 in the development set, and 960 in the test set) from 793 abstracts. It contains one target entity, which is the disease. For the JNPBA dataset, to create the development set, we subdivided the original training set (18,546 sentences) into the training set and the development at 9:1 ratio. The development set was used to monitor the performance of the model and for an early stop to avoid overfitting. Details of each dataset are presented in Table 1.

3.2. Experimental settings

The word embeddings used in this model are initialized using 200-dimensional pre-trained word vectors. Pre-trained word vectors that are publicly available from BioASQ [24] are vectors trained on biomedical articles from PubMed. Words not found in the pre-trained word vectors are initialized as UNK tokens, and words consisting of only digits are replaced with NUM token. The UNK and NUM tokens are randomly initialized and kept fixed during the training. For character embedding, we randomly initialize the character-embedding matrix with a uniform distribution. The sizes of the word- and character- vectors are set to 200-, and 100-, respectively. In the character-level CNN, the sizes of the filter are set to 3, 5, and 7 for the JNLBPA dataset, and 2, 3, and 4 for the NCBI-Disease dataset for optimal performance.

Adam optimizer [14] is employed to optimize parameters with a

Table 1
Dataset details.

Dataset	JNLBPA [13]	NCBI-Disease [7]
Target entity	Protein, DNA, RNA, Cell line, Cell type	Disease
Type	Sentences	Mentions
Train	16,690	5145
Development	1856	787
Test	3856	960

learning rate of 0.001. The decay rate is set as 0.90 for the JNLPBA dataset and 0.95 for the NCBI-Disease dataset. To avoid overfitting, different regularization techniques are applied. The training is performed for 40 epochs; however, it is stopped if no improvement in the performance is achieved for four epochs. In addition, a dropout of 0.5 is applied to the convolution layer and the fully connected layer in the character embedding, as well as to the last hidden state of the bi-LSTM layer. Moreover, L2 regularization is used with the lambda parameter set to 5.0e−4.

3.3. Evaluation metrics

The metrics used for evaluating the performance of the model are the precision, recall, and F1-score, as follows:

$$\text{precision} = \frac{TP}{TP + FP} \tag{9}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{10}$$

$$\text{F1} - \text{score} = \frac{2 * precision * recall}{precision + recall} \tag{11}$$

TP represents a case in which the entities are correctly detected for the words that are labeled as entities, and FP represents a case in which words that are non-entities are detected as entities. Moreover, FN represents a case in which words that are labeled as entities are not detected, and TN represents a case in which words that are labeled as non-entities are correctly detected as non-entities. The precision is the ratio of the number of correctly predicted entities to the total number of identified entities. The recall is the ratio of the number of correctly predicted entities to the total number of entities that exist. The F1-score is the harmonic mean of precision and recall, and it indicates the overall performance of the NER system. Further, we performed the evaluation at the full named entity-level, where the exact boundary match over the entity span of the golden standard corpus.

4. Results and discussion

4.1. Comparison with previous studies

We compared the performance of the proposed model with that of models used in previous studies on two publicly available datasets: JNLPBA and NCBI-Disease. Table 2 presents the results of four competitive bioNER models from the works of Wang et al. [34], Habibi et al. [10], Dang et al. [5], and Gridach [9] evaluated on entity-level matching. In a cross-type multi-task model of Wang et al. [34], which ranked third on JNLPBA and second on NCBI-Disease, a performance improvement is obtained by sharing character- and word-level information between different benchmark datasets. Habibi et al. [10] showed the importance of pre-trained word embedding in the bi-LSTM-CRF model, ranked fourth and third on each dataset, respectively. Dang et al. [5] presented the D3NER model, incorporating linguistic information, such as abbreviations and Part-of-Speech (POS) embedding, which achieved an improved performance on the NCBI-Disease dataset. Finally, the bi-LSTM-CRF model presented by Gridach [9], which employs character-level embeddings, achieved the highest F1-score for the JNLPBA dataset. It has slightly outperformed the F1-score of our model as its F1-score was 0.56% higher. Although our model achieved the second highest performance for the JNLPBA dataset, it achieved the highest F1-score of 86.93% for the NCBI-Disease dataset, outperforming the previously studied models. The results indicate that the proposed method is effective for entity recognition in the biomedical field.

Table 2
Performance evaluation of the proposed model and previous models.

	JNLPBA			NCBI-Disease		
	Precision	Recall	F1	Precision	Recall	F1
Wang et al.	70.91	76.34	73.52	85.86	86.42	86.14
Habibi et al.	71.35	75.74	73.48	86.11	85.49	85.80
Dang et al.	–	–	–	85.03	83.80	84.41
Gridach	**74.16**	77.66	**75.87**	–	–	–
Proposed model	71.89	**79.07**	75.31	**86.75**	**87.11**	**86.93**

Table 3
Effects of Various Types of Proposed Methods in the Model.

		JNLPBA		NCBI-Disease			
#	Model	Precision	Recall	F1	Precision	Recall	F1
(1)	WE	70.00	74.00	71.95	83.43	75.88	79.48
(2)	+ attention	71.07	73.99	72.50	83.94	76.61	80.11
(3)	WE + char(bi-lstm)	70.92	78.86	74.68	85.70	82.85	84.25
(4)	+ attention	71.18	78.98	74.88	85.32	85.76	85.54
(5)	WE + char(cnn)	71.10	78.43	74.59	86.37	84.30	85.32
(6)	+ attention	71.69	77.87	74.65	86.08	84.20	85.13
(7)	WE + char(bi-lstm, cnn)	70.95	78.95	74.74	86.46	85.65	86.06
(8)	+ attention	71.84	78.35	74.95	86.12	85.75	85.94
(9)	+ fully connected network	71.30	78.94	74.93	**86.83**	86.38	86.60
(10)	**+ fully connected network, attention**	**71.89**	**79.07**	**75.31**	86.75	**87.11**	**86.93**

4.2. Contribution of character-level embeddings

To analyze the effects of different types of character embeddings and the effect of the fully connected network that is used to combine character and word vectors, we conducted experiments using models with four different combinations of embeddings. The results are presented in Table 3. In experiment 1, the model used only the word-level embedding (WE), whereas in experiments 3 and 5, the models used embeddings that combine one type of character-level embedding (CNN or bi-LSTM) and the word-level embedding. The results of experiments 3 and 5 significantly outperformed those of experiment 1, indicating that character-level embedding is useful for handling OOV words in NER tasks. Additionally, in experiments 3 and 5, for the JNLPBA dataset, the bi-LSTM model outperformed the CNN model in extracting character-level features, while for the NCBI-Disease dataset, the result was opposite. This indicates that in extracting character-level features, the CNN and bi-LSTM models have similar effects. In experiment 7, we investigated the effect of using two different character-level embeddings combined with the word-level embedding. For both datasets, the proposed model utilizing all three types of embedding (char-bi-lstm, char-cnn, and word) for word representation exhibited the highest performance in experiments 3, 5, and 7, achieving an F1-score of 74.74%, 86.06% for the aforementioned datasets.

4.3. Effects of fully connected network and attention mechanism

To examine the effect of the fully connected network and attention mechanism, we performed six experiments involving addition of components to the models with four different combinations of embeddings.

In experiment 9, embeddings from experiment 7 were fed into the fully connected network to design a new embedding of size 200 that combined three types of embeddings. The results in experiment 9 show that applying the fully connected network to the embedding layer improved F1-score of 0.19%, 0.54% for each dataset compared to experiment 7, which does not apply fully connected network. This indicates that the model effectively captures the most salient features

Table 4
Effects of different types of word embeddings.

	JNLBPA			NCBI-Disease		
	Precision	Recall	F1	Precision	Recall	F1
Random	69.10	76.62	72.67	82.58	78.38	80.43
GloVe	69.52	77.76	73.41	85.12	83.26	84.18
Pubmed-PMC	**71.89**	**79.07**	**75.31**	**86.75**	**87.11**	**86.93**

from each type of embeddings by passing the concatenated embedding through the fully connected network.

In experiment 10, in addition to the fully connected network, the attention mechanism was incorporated into the model. According to the results, the proposed model achieved the highest F1-scores of 75.31% and 86.93%, for the JNLPBA and NCBI-Disease datasets, respectively. An interesting fact was that in experiments 2, 4, 6, and 8, although mechanism had a positive influence overall, the improvements were generally not as significant as those in experiment 10. This demonstrates that the effects of the attention mechanism and the fully connected network are maximized when they are combined in the model with embedding layer that utilizes two different character-level and word-level embeddings.

4.4. Effects of different word embeddings

In the last part of the experiment, we compared the effects of different types of word embedding. We examined three different word-embedding choices: randomly initialized word vectors, and two different types of pre-trained word vectors. The two pre-trained word vectors are Glove embedding, trained on 42 billion tokens from Common Crawl data, and another was trained on biomedical text, as described in Section 3.2. We use the model used in experiment 10 of Table 3. The experimental results are shown in Table 4. The randomly initialized word embeddings exhibited the lowest performance, and the pre-trained word vector, trained on PubMed, had the most positive impact on the performance. This indicates that in bioNER, the use of biomedical related pre-trained embeddings is not only effective for handling OOV words, but also outstanding at capturing the biological meaning of each word token. In fact, 2481 more OOV words were reduced when utilizing biomedical pre-trained embeddings than in the case of using Glove embeddings.

4.5. Error analysis

Table 5 presents example sentences and the types of errors incurred by the proposed model for the JNLPBA test set. The examples in Table 5 are useful for analyzing the limitations of the proposed model and for the further study.

Example 1 shows an error caused by the use of an abbreviations in biomedical terminology. In the test set, “OTFs” is annotated as protein; however, the model predicted “OTFs” as DNA. Thus, the model experiences confusion owing to abbreviations in biomedical terminology. In example 2, the model misrecognized protein entity, “[125I] T3.” It is likely that the model had difficulty recognizing entities consisting of letters, numbers, special characters, and Greek letters. In example 3, the model partially recognized the DNA entity, missing the first word “multiple.” This is a boundary error, where the use of an adjective in an entity name can cause confusion to the model. For boundary errors, it is expected that the attention mechanism helps in recognizing the full name of the entity by capturing the correlation between words in the sentence. Finally, example 4 shows error due to the long and complex structure of biomedical terms.

4.6. Discussion

During the training process, overfitting was a concern due to the small sizes of the JNLPBA and NCBI-Disease datasets. Although we attempted to solve this problem by applying several regularization techniques, it was still challenging to resolve the issue of insufficient data. As an alternative solution to resolve the problem of insufficient data, we can attempt to apply transfer learning for the NCBI-Disease dataset. In future work, we plan to incorporate the knowledge transferring approach to improve bioNER performance and resolve the model limitations revealed by the error analysis.

Moreover, to understand the strength of the proposed model, we have illustrated frequently confirmed correct cases of our model in comparison to the model that utilizes only bi-LSTM or CNN character-level embedding. For case 1 and 2 in Table 6, where entities are composed of multiple tokens, including conjunctions or adjectives, performance of the proposed model is better than the comparable models by predicting the full names of entities. For datasets such as JNLPA with multiple entities included, it is important for the model to accurately identify the absence/presence of the entity type first, and then match the correct boundaries, as in cases 3 and 4. In summary, this elucidates that the proposed model is more efficient in handling the problem of partial match of the corresponding entity in binary datasets with only two classes (e.g. NCBI-Disease dataset), while for the multi-class datasets it performs better in identifying and classifying the correct entity type in comparison to other studies.

5. Conclusion

We presented the model utilizing a deep learning approach for bioNER. The proposed model consists of three designs that contribute to its performance. First, the architecture utilizes two different types of character-level representations extracted from CNN and LSTM neural networks that capture both local and global information from each input token. Second, the model feeds the extracted character- and word-level features to the fully connected network to adaptively learn the combination of each representation by training the best features from each embedding. Finally, the attention mechanism is employed to effectively find the relations between words, which enables efficient recognition of entities. We evaluated the performance of the proposed model on two benchmark datasets, JNLPBA and NCBI-Disease, and compared it with that of other competitive models, as shown in Table 2. Our model achieved the second highest F1-score for the JNPBA dataset, and the highest F1-score for the NCBI-Disease dataset. In summary, the experimental results indicate that the proposed model is effective in designing high-level features of embeddings that capture meaningful information from entities especially in the biomedical domain. Additionally, the result demonstrates that the proposed model is a competitive NER system for extracting biomedical entities from various resources.

CRediT authorship contribution statement

Minsoo Cho: Conceptualization, Investigation, Methodology, Writing - original draft, Writing - review & editing, Software. **Jihwan Ha:** Conceptualization, Writing - review & editing, Validation. **Chihyun Park:** Writing - review & editing, Validation. **Sanghyun Park:** Conceptualization, Funding acquisition, Project administration, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 5
Examples of Error Caused by the Proposed Model for the JNLPBA Dataset.

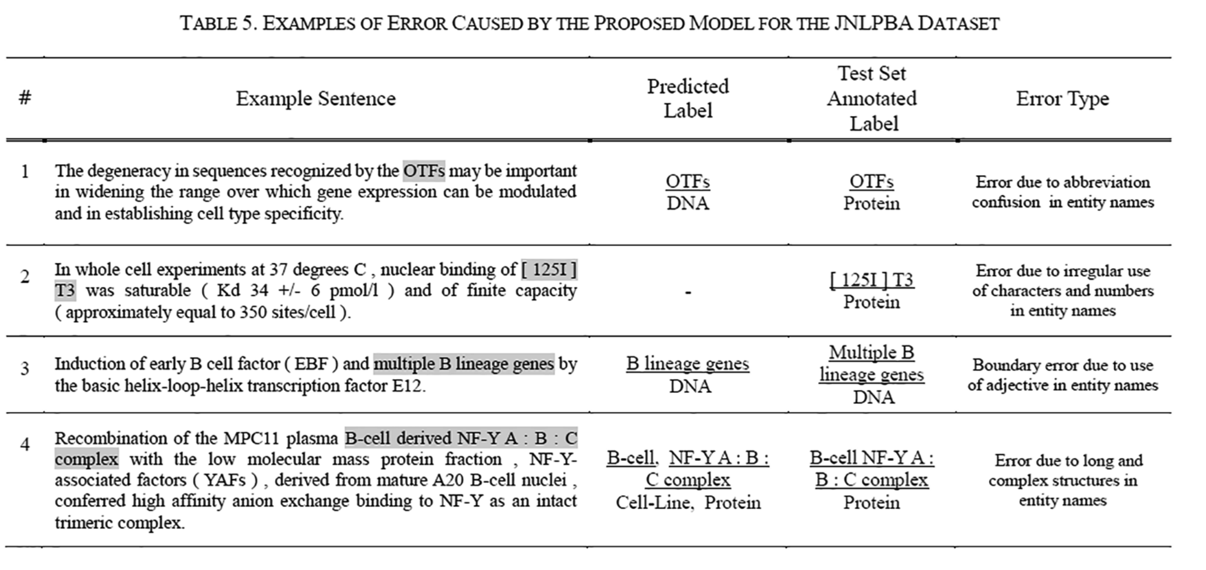


TABLE 5. EXAMPLES OF ERROR CAUSED BY THE PROPOSED MODEL FOR THE JNLPBA DATASET

#	Example Sentence	Predicted Label	Test Set Annotated Label	Error Type
1	The degeneracy in sequences recognized by the OTFs may be important in widening the range over which gene expression can be modulated and in establishing cell type specificity.	OTFs DNA	OTFs Protein	Error due to abbreviation confusion in entity names
2	In whole cell experiments at 37 degrees C , nuclear binding of [125I] T3 was saturable (Kd 34 +/- 6 pmol/l) and of finite capacity (approximately equal to 350 sites/cell).	-	[125I] T3 Protein	Error due to irregular use of characters and numbers in entity names
3	Induction of early B cell factor (EBF) and multiple B lineage genes by the basic helix-loop-helix transcription factor E12.	B lineage genes DNA	Multiple B lineage genes DNA	Boundary error due to use of adjective in entity names
4	Recombination of the MPC11 plasma B-cell derived NF-Y A : B : C complex with the low molecular mass protein fraction , NF-Y-associated factors (YAFs) , derived from mature A20 B-cell nuclei , conferred high affinity anion exchange binding to NF-Y as an intact trimeric complex.	B-cell, NF-Y A : B : C complex Cell-Line, Protein	B-cell NF-Y A : B : C complex Protein	Error due to long and complex structures in entity names

Table 6
Case study of different models for the JNLPBA- and NCBI- datasets.

#	Model	NCBI-Disease
1	WE + char(cnn)	… with congenital cataracts, with autosomal dominant keratitis, and with isolated **foveal hypoplasia.**
	WE + char(bi-lstm)	… with congenital cataracts, with autosomal dominant keratitis, and with isolated **foveal hypoplasia.**
	Proposed model	… with congenital cataracts, with autosomal dominant keratitis, and with **isolated foveal hypoplasia.**
2	WE + char(cnn)	Hemochromatosis, the **inherited disorder** of iron metabolism, leads, if untreated, to progressive iron overload …
	WE + char(bi-lstm)	Hemochromatosis, the **inherited disorder** of iron metabolism, leads, if untreated, to progressive iron overload …
	Proposed model	Hemochromatosis, the **inherited disorder of iron metabolism**, leads, if untreated, to progressive iron overload …
#	Model	JNLPBA
3	WE + char(cnn)	… the number of adhering leukocytes in respect to native albumin used as control (110 + /-16 versus 66 + /-7, P < 0.01).
	WE + char(bi-lstm)	…. the number of adhering leukocytes in respect to native albumin used as control (110 + /-16 versus 66 + /-7, P < 0.01).
	Proposed model	… the number of adhering leukocytes in respect to **native albumin** used as control (110 + /-16 versus 66 + /-7, P < 0.01).
4	WE + char(cnn)	… NFAT gene expression in **human interleukin-2-dependent T lymphoblasts** stimulated via T-cell receptor. (Cell-type)
	WE + char(bi-lstm)	… NFAT gene expression in **human interleukin-2-dependent T lymphoblasts** stimulated via T-cell receptor. (Cell-type)
	Proposed model	… NFAT gene expression in **human interleukin-2-dependent T lymphoblasts** stimulated via T-cell receptor. (Cell-line)

* The underlined part of each sentence corresponds to the biomedical entities that each model has predicted. The table illustrates four cases where the proposed model has correctly predicted the labels, while the other two models have not correctly predicted the labels.

Acknowledgment

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the SW Starlab support program (IITP-2017-0-00477) supervised by the IITP (Institute for Information & communications Technology Promotion).

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jbi.2020.103381.

References

[1] D. Campos, S. Matos, J.L. Oliveira, Biomedical named entity recognition: a survey of machine-learning tools, Theor. Appl. Adv. Text Min. (2012).
[2] W.W. Chapman, et al., A simple algorithm for identifying negated findings and diseases in discharge summaries, J. Biomed. Inform. 34 (5) (2001) 301–310.
[3] J.P.C. Chiu, E. Nichols, Named entity recognition with bidirectional LSTM-CNNs, arXiv preprint arXiv:1511.08308, 2015.
[4] K. Cho et al., Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv preprint arXiv: 1406.1078, 2014.
[5] T.H. Dang, H.Q. Le, T.M. Nguyen, S.T. Vu, D3NER: Biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information, Bioinformatics 1 (2018).
[6] J. Devlin, M. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805, 2018.
[7] R.I. Doğan, R. Leaman, Z. Lu, NCBI disease corpus: a resource for disease name recognition and concept normalization, J. Biomed. Inform. 47 (2014) 1–10.
[8] C. Friedman, et al., GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles, ISMB (supplement of bioinformatics), (2001).
[9] M. Gridach, Character-level neural network for biomedical named entity recognition, J. Biomed. Inform. 70 (June 2017) 85–91.
[10] M. Habibi, L. Weber, M. Neves, D.L. Wiegandt, U. Leser, Deep learning with word embeddings improves biomedical named entity recognition, Bioinformatics 33 (July 2017) i37–i48.
[11] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780.

[12] N. Kanya, T. Ravi, Machine learning based biomedical named entity recognition, IET Chennai Fourth International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2013), 2013 pp. 380 – 384, 2013.

[13] J.D. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, N. Collier, Introduction to the bio-entity recognition task at JNLPBA, Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, Association for Computational Linguistics, 2004, pp. 70–75.

[14] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

[15] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.

[16] G. Kumaran, J. Allan, Text classification and named entities for new event detection, Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2004, pp. 297–304.

[17] J. Lafferty, A. McCallum, F.C.N Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, 2001.

[18] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, arXiv preprint arXiv:1603.01360, 2016.

[19] J. Lee et al. BioBERT: pre-trained biomedical language representation model for biomedical text mining, arXiv preprint arXiv:1901.08746, 2019.

[20] U. Leser, J.J. Hakenberg, What makes a gene name? Named entity recognition in the biomedical literature, Briefings Bioinformatics 6 (2005) 357–369.

[21] L. Luo, et al., An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition, Bioinformatics 34 (2017) 1381–1388.

[22] M.T. Luong, H. Pham, C.D. Manning, Effective approaches to attention-based neural machine translation, arXiv preprint arXiv:1508.04025, 2015.

[23] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional lstm-cnns-crf, arXiv preprint arXiv:1603.01354, 2016.

[24] Y. Mao, C.H. Wei, Z. Lu, NCBI at the 2014 BioASQ Challenge Task: Large-scale Biomedical Semantic Indexing and Question Answering, CLEF, (Working Notes), 2014.

[25] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.

[26] D. Mollá, M. Van Zaanen, D. Smith, Named entity recognition for question answering, Proceedings of the 2006 Australasian Language Technology Workshop, Sancta Sophia College, Sydney, vol. 4, 2006, pp. 51–58 ISSN 1834-7037.

[27] N. Ponomareva, P. Rosso, F. Pla, A. Molina, Conditional random fields vs. hidden Markov models in a biomedical named entity recognition task, Proc. of Int. Conf. Recent Advances in Natural Language Processing, RANLP, 2007, pp. 479–483.

[28] M. Rei, G.K. Crichton, S. Pyysalo, Attending to characters in neural sequence labeling models, arXiv preprint arXiv:1611.04361, 2016.

[29] D.S. Sachan et al., Effective use of bidirectional language modeling for transfer learning in biomedical named entity recognition, arXiv preprint arXiv:1711.07908, 2017.

[30] B. Settles, Biomedical named entity recognition using conditional random fields and rich feature sets, Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications, 2004, pp. 104–107.

[31] M. Song, W.C. Kim, D. Lee, G.E. Heo, K.Y. Kang, PKDE4J: Entity and relation extraction for public knowledge discovery, J. Biomed. Inform. 57 (October 2015) 320–332.

[32] R.T.H. Tsai, et al., NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition, BMC Bioinformatics 7 (5) (2006) S11 BioMed Central.

[33] Y. Tsuruoka, J.I. Tsujii, Improving the performance of dictionary-based approaches in protein name recognition, J. Biomed. Inform. 37 (May 2004) 461–470.

[34] X. Wang et al., Cross-type Biomedical Named Entity Recognition with Deep Multi-Task Learning, arXiv preprint arXiv:1801.09851, 2018.

[35] K. Xu, et al., Show, attend and tell: Neural image caption generation with visual attention, International Conference on Machine Learning, 2015, pp. 2048–2057.

[36] W. Yoon, S.H. Chan, J. Lee, J. Kang, CollaboNet: collaboration of deep neural networks for biomedical named entity recognition, BMC Bioinf. 20 (10) (2019) 249.

[37] S. Zhao, T. Liu, T., S. Zhao, F. Wang, A Neural Multi-Task Learning Framework to Jointly Model Medical Named Entity Recognition and Normalization, CoRR, 2018.