Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8894760, 13 pages
https://doi.org/10.1155/2020/8894760

*Research Article*

# A Multichannel Biomedical Named Entity Recognition Model Based on Multitask Learning and Contextualized Word Representations

**Hao Wei, Mingyuan Gao, Ai Zhou, Fei Chen, Wen Qu, Yijia Zhang, and Mingyu Lu**

*School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China*

Correspondence should be addressed to Mingyu Lu; lumingyu@dlmu.edu.cn

Received 24 May 2020; Revised 15 June 2020; Accepted 30 June 2020; Published 10 August 2020

Academic Editor: Yin Zhang



As the biomedical literature increases exponentially, biomedical named entity recognition (BNER) has become an important task in biomedical information extraction. In the previous studies based on deep learning, pretrained word embedding becomes an indispensable part of the neural network models, effectively improving their performance. However, the biomedical literature typically contains numerous polysemous and ambiguous words. Using fixed pretrained word representations is not appropriate. Therefore, this paper adopts the pretrained embeddings from language models (ELMo) to generate dynamic word embeddings according to context. In addition, in order to avoid the problem of insufficient training data in specific fields and introduce richer input representations, we propose a multitask learning multichannel bidirectional gated recurrent unit (BiGRU) model. Multiple feature representations (e.g., word-level, contextualized word-level, character-level) are, respectively, or collectively fed into the different channels. Manual participation and feature engineering can be avoided through automatic capturing features in BiGRU. In merge layer, multiple methods are designed to integrate the outputs of multichannel BiGRU. We combine BiGRU with the conditional random field (CRF) to address labels' dependence in sequence labeling. Moreover, we introduce the auxiliary corpora with same entity types for the main corpora to be evaluated in multitask learning framework, then train our model on these separate corpora and share parameters with each other. Our model obtains promising results on the JNLPBA and NCBI-disease corpora, with F1-scores of 76.0% and 88.7%, respectively. The latter achieves the best performance among reported existing feature-based models.

## 1. Introduction

Named entity recognition (NER) aims to identify and extract specific entities (persons, places, organizations, and so on) from massive unstructured text data, which becomes a primary task for information extraction, text analysis, text mining, etc. Similarly, how to effectively extract and obtain valuable information has become a serious challenge for researchers in the biomedical field. Biomedical named entity recognition (BNER) is an indispensable step for this above challenge. The biomedical entities consist of genes, proteins, diseases, drugs, chemicals, and so on.

In the past, conventional machine learning methods were widely used for NER, such as support vector machine (SVM), conditional random field (CRF), and maximum entropy model (MEM). Finkel et al. [1] combined distant resources and additional features to identify the biomedical entities. Tsuruoka et al. [2] employed MEM to develop a BNER system named GENIA Tagger. ABNER [3] was a biomedical entities extraction system based on CRF. Chang et al. [4] adopted the biomedical word embeddings as external features to improve the performance of CRF significantly. Liao et al. [5] adopted the Skip-Chain CRF model to recognize entities, which effectively captured the features of the distant context. Tang et al. [6] used a CRF model with three different types of word representations to identify biological entities. According to the above studies, CRF had became the mainstream model in BNER [7]. Nevertheless, feature engineering is an essential element of the conventional machine learning methods. They must manually

design complex templates that require not only domain knowledge but also time-consuming.

Driven by artificial intelligence and pattern recognition, some labor-saving and advanced technologies have been developed in natural language processing, computer vision, and other emerging fields [8–17]. For example, deep learning can obviously address the expensive cost of feature engineering. The widely employed neural networks include convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and gated recurrent unit networks (GRUs). Yao et al. [18] first built a multilayer neural network to obtain the biomedical word embeddings on large-scale corpora. To extract disease and chemical entities, Zhao et al. [19] constructed a CNN model. In this work, BNER was seen as text classification, and a multilabel mechanism was designed to obtain contiguous labels. Zhu et al. [20] adopted a CNN structure in BNER with $n$-gram local character and word embeddings. The GRAM-CNN obtained the best performance (F1-score: 87.3) among the single-task models on the NCBI-disease corpus. Li et al. [21] made connections between the twin word embeddings and sentence vectors. Furthermore, they adopted the bidirectional LSTM (BiLSTM) to identify biomedical entities and significantly improved the performance. Limsopatham et al. [22] proposed an end-to-end model based on BiLSTM and orthographic features. It was designed to improve the extraction of complex biomedical terms. SBLC was developed by Xu et al. [23] based on word embeddings and BiLSTM-CRF structure. Dang et al. [24] also proposed the BNER model based on the BiLSTM-CRF structure and adopted various fine-tuned linguistic embeddings. The model showed high performance on multiple corpora. Lyu et al. [25] adopted the BiLSTM-RNN model and combined the biomedical word embeddings with character embeddings to recognize entities. In addition, some studies based on multitask learning and transfer learning were widely used in BNER and had achieved competitive performance. Wang et al. [26] jointly trained different types of entities in multiple data sets and shared both word and character representations among relevant entities. The multitask model achieved promising performance on 15 biomedical corpora. Yoon et al. [27] proposed a multitask framework termed CollaboNet. It connected multiple submodels trained on different corpora. The large performance gains come from taking turns training the target and collaborator submodels. Sachan et al. [28] designed a pretrained BiLSTM model. They first trained a language model of the same structure on the unlabeled corpora and then updated the initialization parameters of the BNER model based on transfer learning. It does not only substantially improved the performance but also alleviated the lack of high-quality labeled training data.

From the above studies, word embeddings can be seen to have become indispensable representations. They can effectively represent the semantic features of the original text sequences. But biomedical entities' naming rules are vague. There are many polysemous and ambiguous words in the biomedical literature. For example, in "This cohort underwent follow-up for cancer incidence through the Finnish cancer registry to the end of 1995.", the first "cancer" means disease and the second is an institution. In addition, it is difficult to address the lack of sufficient training samples in specific fields. These issues also result that the biomedical entities are more complex to recognize than the general field. Because the traditional fixed word embeddings cannot accurately represent polysemous and ambiguous words in the biomedical literature, the language models pretrained on a large number of unlabeled open corpora have drawn more and more attention. The contextualized word embeddings generated by them can optimize the feature representations of the polysemous and ambiguous words. In the general field, Peters et al. [29] designed a feature-based language model named ELMo, which consists of a bidirectional LSTM. This pretrained language model achieves state-of-the-art performance in multiple downstream tasks.

We aim to optimize the representations of polysemic words and ambiguous words in biomedical sequences and make the model fully capture richer features. This paper proposes a multitask learning multichannel BiGRU-CRF model with feature-based contextualized word representations. The main contributions of this paper are as follows.

1) We propose a multichannel BiGRU-CRF model. Three kinds of feature representations based on the biomedical pretrained dictionary, ELMo, and CNN are generated, including word-level, contextualized word-level, and character-level representations. These representations are separated or combined as inputs simultaneously, and each set of inputs is fed into a BiGRU-CRF model as a single channel. In merge layer, multiple methods are designed to integrate the outputs of multichannel BiGRU.

2) In order to address the lack of sufficient training data in specific fields, we adopt multitask learning strategy, employing auxiliary corpora to provide richer training samples and relevant information for the main corpora to be evaluated.

3) The multitask learning multichannel BiGRU-CRF model clearly strengthens the capability of recognizing entities without any artificial participation. It obtains the competitive results on the JNLPBA and NCBI-disease corpora.

The rest of this paper is divided into the following four sections. Section 2 describes the methods. Section 3 shows the experimental settings. Section 4 reports the evaluative results in a detailed manner. Section 5 provides the conclusion.

## 2. Methods

Figure 1 shows the multitask learning multichannel BiGRU-CRF framework. The framework is divided into five parts: input layer, embedding layer, BiGRU, merge layer, and CRF layer, where the input layer represents the original sentence in corpora. First, the three feature representations are obtained through biomedical pretrained dictionary, CNN, and ELMo language model, respectively. Then, the multichannel BiGRU is used to capture features. $\vec{h}_{0-6}$ denotes the forward single-channel GRU, and $\overleftarrow{h}_{0-6}$ denotes the backward single-channel GRU, respectively. Next, we integrate the output of each channel in the merge layer. Finally, the labels are parsed by CRF. This section describes the remaining four parts in detail.

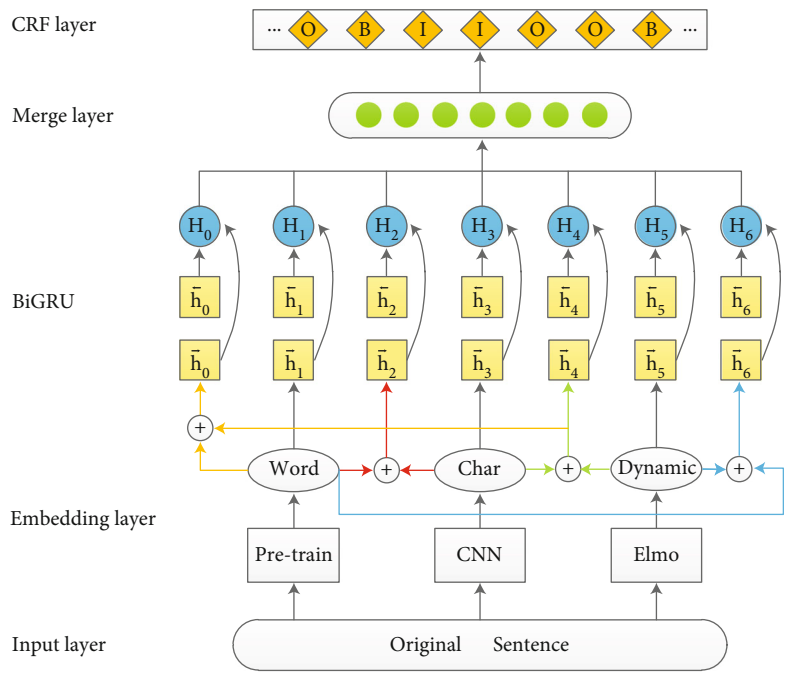


FIGURE 1: Multichannel BiGRU-CRF architecture. It consists of 5 parts: input layer, embedding layer, BiGRU, merge layer, and CRF layer. The red, green, and blue lines, respectively, represent the channels after the concatenate operation of two representations, and the yellow line represents the channel after the concatenate operation of all three representations. *H*0-6, respectively, denotes the bidirectional single-channel GRU. Each channel is independent, which avoids information redundancy.

*2.1. Embedding Layer.* To ensure the maximum coverage of the input information, the pretrained word embeddings, contextualized word embeddings, and character embeddings are used for the input layer for feature representations.

*2.1.1. Pretrained Word Embedding.* We represent the text sequence with word embeddings. They map words to dense vectors according to semantic relevance. The word embedding method addresses the lack of curse of dimensionality compared with the conventional one-hot method. With the development of natural language processing, word embeddings have become the most important input feature representations. The widely adopted word embedding computing tools include Word2Vec [30] and GloVe [31].

Previous biomedical studies have provided related open source word embeddings pretrained on large-scale unlabeled corpora. We initialize the word embeddings by a "look up" operation. Inspired by Quan et al. [32], this paper adopts the word embeddings pretrained on *PMC* and *PubMed* biomedical corpora.

*2.1.2. Contextualized Word Embedding.* This paper directly transfers the pretrained ELMo language model proposed by Peters et al. [29] to obtain the contextualized word embeddings. The main motivation is that the contextualized word representations should be able to contain rich syntactic and semantic information. The conventional word embeddings (e.g., word2vec) are context-independent, and ELMo can generate dynamic word embeddings based on context. We adopt the 2-layer ELMo to obtain the contextualized word representations as part of the multichannel BiGRU-CRF model's input, which is shown in Figure 2. ELMo consists of a bidirectional LSTM language model. The objective function is to compute the maximum likelihood of the two submodels. For $k-th$ word, a set of contextualized word representations can be computed by ELMo as follows:

$$\begin{aligned} ELMo_k &= \sum_{j=0}^{L} w h_{k,j}^{LM} \\ R_k &= \left\{ x_k^{LM}, h_{k,j}^{L\vec{M}}, h_{k,j}^{L\overleftarrow{M}} \right\}, j = \{1, \cdots, L\} \\ R_k &= \left\{ h_{k,j}^{LM} \right\}, j = \{0, \cdots, L\} \end{aligned} \tag{1}$$

where $x_k^{LM}$ denotes the original embeddings layer. $h_{k,j}^{L\vec{M}}$ and $h_{k,j}^{L\overleftarrow{M}}$ denote the forward and backward LSTM layer, respectively. $w$ denotes the softmax-normalized weights, and $L$ denotes the number of layers. ELMo generates word representations based on the above formula, which is summing each hidden state of the bidirectional language model. They can be directly concatenated with other feature inputs. The contextualized word embeddings not only reflect the complex semantics and grammar features but also accurately adapt to different contexts.

*2.1.3. Character Embedding.* Character representations refer to morphological information by capturing it from all characters that make up a word. Combining them with other

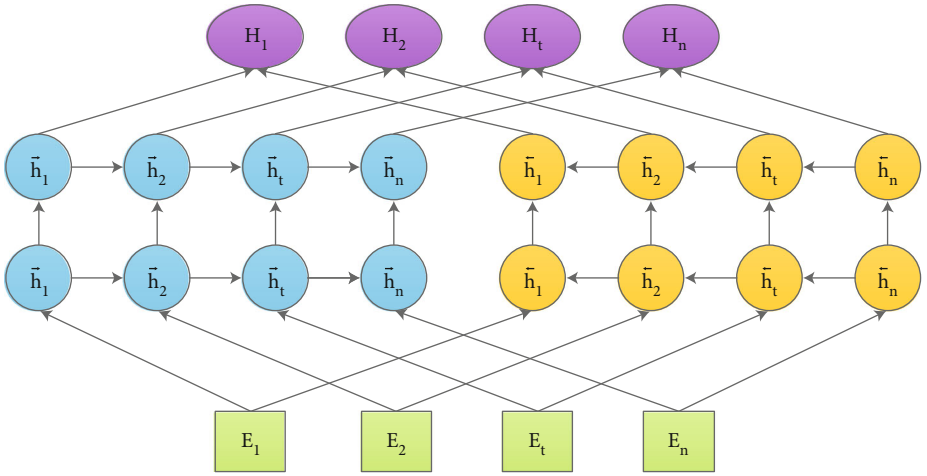


Figure 2: The framework of ELMo.

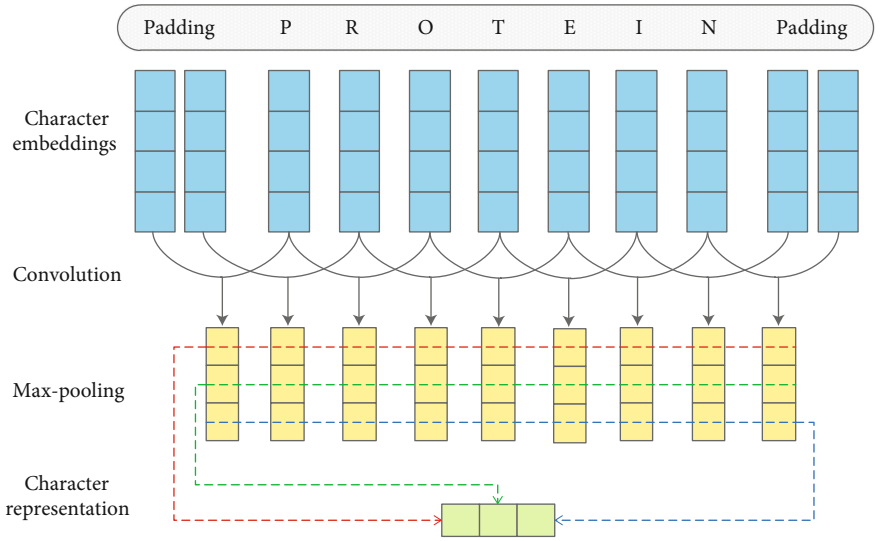


Figure 3: The CNN model framework.

feature representations can better describe the morphological features of a word [33, 34]. Previous studies have shown the effectiveness of character representations in NER. This paper adopts CNN to compute the character vectors of words in biomedical sequences. The structure of CNN is shown in Figure 3, including the original character embeddings by random initialization, convolutional layer, and pooling layer. First, the words' embeddings matrix consists of each character embeddings. A padding operation for words of different lengths is performed. Then, the local features of the initialized character embeddings matrix are captured by a convolution operation. Finally, the character representations are obtained by performing a max-pooling operation.

*2.2. Multichannel BiGRU.* Recently, to solve the gradient explosion or gradient disappearance, a variety of improved models based on RNN have been proposed, such as LSTM [35] and GRU [36–38]. They capture distant information and address the gradient disappearance or gradient explosion by designing the memory units and gate mechanisms. Therefore, the above improved models have become the major option for sequence labeling such as BNER. The difference between LSTM and GRU is the structure of gate mechanisms. GRU maintains the performance of LSTM while making the gate structures simpler [39, 40]. Because we need to train multiple identical networks at the same time, this paper adopts GRU with lower computational complexity. Figure 4 shows the GRU units. The relevant formulas are as follows.

$$\begin{aligned} z_t &= \sigma(W_z[h_{t-1}, x_t]) \\ r_t &= \sigma(W_r[h_{t-1}, x_t]) \\ h_t^{\sim} &= \tanh\left(W[r_t * h_{t-1}, x_t]\right) \\ h_t &= (1 - z_t) * h_{t-1} + z_t * h_t^{\sim} \end{aligned} \tag{2}$$

where $\sigma$ denotes the *sigmod* function. $z_t$ and $r_t$ denote the update and reset gate. $x_t$ denotes the feature vectors. $W$ denotes the weights of the gate mechanism. $h_t^{\sim}$ denotes the

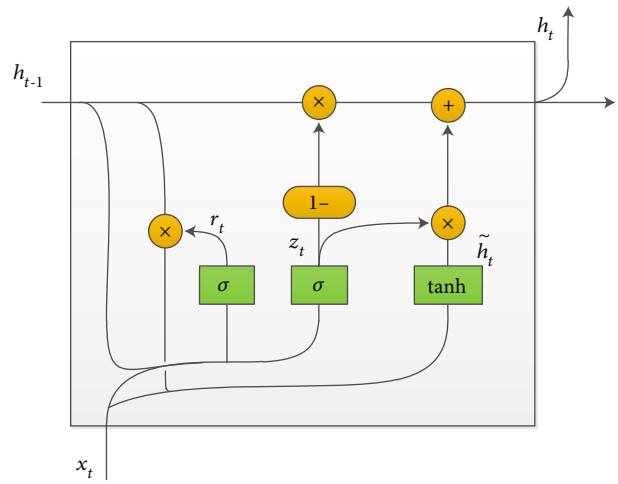


Figure 4: The unit of GRU.

current state. tanh denotes the hyperbolic tangent function. $h_t$ denotes the final output.

However, GRU only considers the forward information of texts and ignores the backward information, which also contains important features. The bidirectional GRU is employed in our model because of this issue. The BiGRU model captures different bidirectional feature representations in each sequence. Then, it obtains the complete representations by connecting them. BiGRU can capture the bidirectional representations and hidden features. In our model, we propose a multichannel BiGRU to obtain the richer representations. The multichannel mechanism aims to feed different kinds of input representations into corresponding multiple independent and same network structures. Each channel uses a separate BiGRU to capture features, which does not cause interference between the channels and can extract information more adequately. A total of 7 channels are designed to capture features of different representations, as follows.

(1) 1st channel: pretrained word embeddings ⊕ contextualized word embeddings ⊕ character embeddings

(2) 2nd channel: pretrained word embeddings

(3) 3rd channel: pretrained word embeddings ⊕ character embeddings

(4) 4th channel: character embeddings

(5) 5th channel: contextualized word embeddings ⊕ character embeddings

(6) 6th channel: contextualized word embeddings

(7) 7th channel: pretrained word embeddings ⊕ character embeddings

where ⊕ denotes the concatenate operation.

*2.3. Merge Layer.* The purpose of using the merge layer is to integrate the outputs of multiple channels from BiGRU. A good merge scheme can effectively integrate the potential valuable information in multichannel BiGRU. As shown in Figure 1, the multichannel BiGRU is adopted to capture features from different representations. Let $H'$ denotes the multichannel BiGRU's output. For a given text sequence $S = \{s_1, s_2, \cdots, s_m\}$, $m$ denotes the length of the sequence, and $u$ denotes the number of BiGRU units. We design four merge methods: addition, connection, unit-level attention, and channel-level attention.

1) Addition. This method additively integrates the output of each channel, and each single BiGRU does not interfere with others when capturing features. It can be obtained as follows:

$$\begin{aligned} H_i &= \left[\overrightarrow{h_i} \oplus \overleftarrow{h_i}\right] \\ H' &= H_w + H_e + H_c + H_{\text{we}} + H_{wc} + H_{ec} + H_{wec} \end{aligned} \tag{3}$$

where + denotes element-wise addition, $H' \in m \times u$. $H_i$ denotes the single-channel BiGRU's output, $w$, $e$, and $c$, respectively, denote the pretrained word embeddings, the contextualized word embeddings from ELMo, and the character embeddings from CNN.

2) Connection. This method directly performs the concatenate operation on the single-channel BiGRU's output. It can be obtained as follows:

$$\begin{aligned} H_i &= \left[\overrightarrow{h_i} \oplus \overleftarrow{h_i}\right] \\ H' &= H_w \oplus H_e \oplus H_c \oplus H_{\text{we}} \oplus H_{wc} \oplus H_{ec} \oplus H_{wec} \end{aligned} \tag{4}$$

where ⊕ denotes the concatenate operation, $H' \in m \times 7u$. $w$, $e$, and $c$, respectively, denote the 3 different embeddings.

3) Unit-level attention. This method adopts the multihead self-attention mechanism to redistribute the weights of units in BiGRU. It can be obtained as follows:

$$\begin{aligned} H_i &= \left[\overrightarrow{h_i} \oplus \overleftarrow{h_i}\right] \\ \alpha &= Softmax\left(\frac{QK^T}{\sqrt{u}}\right) \\ head_i &= \sum_m \alpha V \\ MH(Q, K, V)_i &= (head_1 \oplus \cdots \oplus head_H) \\ H' &= \sum_{i=1}^{n} MH_i \end{aligned} \tag{5}$$

where ⊕ denotes the concatenate operation. $H_i$ denotes the single-channel BiGRU's output, $Q, K, V \in m \times (u/H)$, $MH_i \in m \times u$, $H' \in m \times u$.

4) Channel-level attention. This method first connects the feature representations of all channels, then computes the weights of each channel and finally integrates them. It

can be obtained as follows:

$$\begin{aligned} H_i &= \left[\vec{h_i} \oplus \overleftarrow{h_i}\right] \\ H &= H_w \oplus H_e \oplus H_c \oplus H_{\text{we}} \oplus H_{wc} \oplus H_{ec} \oplus H_{wec} \\ \alpha_i &= \frac{\exp\,(e_i)}{\sum_{k=1}^{n} \exp(e_k)} \\ e_i &= \tanh\,\left(W^T H + b\right) \\ H' &= H \otimes \alpha_i \end{aligned} \tag{6}$$

where $\otimes$ denotes matrix multiplication, $H \in m \times u \times 7$, $H' \in m \times u$.

*2.4. CRF Layer.* After the representations information is output by BiGRU, the conventional decision function computes the prediction labels $Y$. However, the output sequence labels have strong dependence in BNER. For example, in the *BIO* labeling scheme, the previous label of "B-disease" cannot be "I-disease". The conventional decision function is insufficient to address the above issue effectively.

In our model, CRF [41] is employed after the merge layer; hence, the dependence between the output labels can be effectively considered. For sentence $X = \{x_1, x_2, \cdots, x_m\}$, it is input into BiGRU. $P$ denotes the probability which is output from merge lager, $P \in m \times n$. $m$ denotes the sequences, and $n$ denotes the labels. $p_{ij}$ denotes the $j-th$ label probability of the $i-th$ token. $Y$ denotes the prediction labels, where $Y = \{y_1, y_2, \cdots, y_m\}$. Its probability can be obtained as:

$$P(X, Y) = \sum_{i=0}^{m} F_{y_i, y_{i+1}} + \sum_{i=1}^{m} P_{i, y_i} \tag{7}$$

where $F$ denotes the transfer matrix. $F_{y_i, y_{i+1}}$ denotes the transition probability from $y_i$ to $y_{i+1}$. The probability of all prediction labels $Y$ by decision function can be computed as follows:

$$P(Y \mid X) = \frac{exp^{P(X,Y)}}{\sum_{Y^{\sim} \in Y_X} exp^{P(X, Y^{\sim})}} \tag{8}$$

$Y^{\sim}$ denotes the truth labels.

The likelihood function is:

$$\log\,(P(Y \mid X)) = P(X, Y) - \log\left(\sum_{Y^{\sim} \in Y_X} exp^{P(X, Y^{\sim})}\right) \tag{9}$$

$Y_X$ denotes all legal label sequences. The final prediction label sequence with the maximum probability can be gained as follows:

$$Y^* = argmax_{Y^{\sim} \in Y_X} P(X, Y^{\sim}) \tag{10}$$

*2.5. Multitask Learning.* In order to provide more training data and value information for our model, we adopt the multitask learning strategy. The basic idea of multitask learning is to learn multiple tasks at the same time and use related information between tasks to improve model performance. The neural network-based multitask learning method mainly adopts a parameter sharing learning mode to learn a shared representation for multiple tasks. In this paper, we introduce two auxiliary corpora with the same entity types for the main corpora to be evaluated, then train the multichannel BiGRU-CRF model on these separate corpora and share parameters with each other.

Given a set of training corpus $n$, $n \in \{1, \cdots, n\}$. $X_i$ and $Y_i$ represent the samples and corresponding prediction labels in each corpus, respectively. The loss function $L$ of the model based on multitask learning is as follows:

$$\begin{aligned} L &= \sum_{i=1}^{n} \alpha_i L_i \\ &= \sum_{i=1}^{n} \alpha_i \log\,(P(Y_i \mid X_i)) \\ &= \sum_{i=1}^{n} \alpha_i \left(P(X_i, Y_i) - \log\left(\sum_{Y_i^{\sim} \in Y_X} exp^{P(X_i, Y_i^{\sim})}\right)\right) \end{aligned} \tag{11}$$

where $\alpha_i$ is a hyperparameter that reflects the weight of each corpus. It represents the contribution and importance of all participating corpora in the whole. When we can obtain that $\alpha$ is 1 through a large number of experiments, that is, when weights are not distinguished, the model reaches the highest performance, which is also consistent with the conclusion of Wang et al. [26].

This paper adopts the fully-shared mode, which means that all parameters of the model are completely shared except that a corresponding output layer is set for each corpus. We provide an auxiliary corpus for the main corpus. The fully shared multichannel BiGRU can capture shared feature representations for multiple corpora, which are fed into their respective output layers to generate prediction sequences.

## 3. Experimental Settings

In this section, the experimental settings are reported clearly, including optimizer and regularization, hyperparameters, corpora, and evaluation measures.

*3.1. Optimizer and Regularization.* Adam [42] (Adaptive Moment Estimation) is adopted as the optimizer of our model during training. It is an adaptive optimization method that dynamically updates the learning rate by computing the gradient's 1st moment estimate and 2nd moment estimate. Each adjusted learning rate is limited to a clear range, which ensures that the parameters are steadily updated.

We use dropout during model training to prevent overfitting. Dropout [43] is designed to randomly filter some hidden layer nodes according to the preset dropout rate so that they do not participate in the back propagation to update

parameters. The above operations can effectively prevent overfitting. They make the model more generalized.

*3.2. Hyperparameters.* Table 1 reports the experimental hyperparameter settings. The dimension based on the pretrained word embeddings, character embeddings, and contextualized word embeddings is set to 200, 30, and 1024, respectively. We adopt the Adam to optimize our model during training. The dimension of GRU units is 100, and the dropout rate is 0.5. We set learning rate as 0.001, and the batch size is 32. In this paper, *BIO* labeling schema is employed to preprocess the original samples. *B* denotes the first token of entities in samples. *I* denotes the token located in entities. *O* denotes a token not belonging to entities.

*3.3. Corpora.* JNLPBA [44] and NCBI-disease [45] are our experimental main corpora. They are representative biomedical corpora of both multi and single classification. JNLPBA contains 5 types of entity: DNA, RNA, cell type, cell line, and protein. Training sets contain 2000 Medline abstracts, and test sets contain 404 Medline abstracts. The NCBI-disease corpus consists of 793 Medline abstracts, of which 593, 100, and 100, are used as training set, development set, and test set, respectively. It labels the disease name and the corresponding disease concept ID (the concept ID can be mapped to the ID in the MeSH or OMIM database). In addition, in the multitask learning framework, we use two other corpora as auxiliary data sets, namely BC2GM [46] and BC5CDR-disease [47]; the entity types contained in these two corpora are consistent with the main corpora. Table 2 provides the details of the above corpora.

*3.4. Evaluation Measures.* To evaluate the performance of our method, we adopt three conventional evaluation measures: precision ($P$), recall ($R$), and F1-score ($F1$). The calculation formulas are as follows:

$$P = \frac{TP}{TP + FP}$$
$$R = \frac{TP}{TP + FN} \quad (12)$$
$$F1 = \frac{2 * P * R}{P + R}$$

where $TP$ denotes the number of true positive samples. $TN$ denotes the number of true negative samples. $FP$ denotes the number of false-positive samples. $FN$ denotes the number of false-negative samples.

## 4. Results and Discussions

The described multitask learning multichannel BiGRU-CRF model is evaluated on NCBI-disease and JNLPBA. They are representative biomedical corpora of both single and multi-classification. We first compare the performance of each merge method and feature representations, as shown in Tables 3 and 4. Then, we evaluate the setting of hyperparameter values including the GRU dimension, optimizers, and dropout, as shown in Tables 5, 6, and 7. From Table 8, the effect of the CRF layer in our architecture is shown by an experiment. From Table 9, the effect of the multitask learning strategy is shown by an experiment. Lastly, the experiment compares the performance of multichannel BiGRU with some existing feature-based methods in BNER.

Table 1: Experimental parameter settings.

| Hyperparameter | Value |
|---|---|
| Word dim | 200 |
| Char dim | 30 |
| ELMo dim | 1024 |
| GRU dim | 100 |
| Head | 8 |
| $\alpha_i$ | 1 |
| Dropout rate | 0.5 |
| Initial learning rate | 0.001 |
| Optimizer | Adam |
| Batch size | 32 |
| Labeling schema | BIO |

*4.1. Performance Comparison of Merge Methods.* The merge methods affect the performance of capturing features. In the merge layer, inappropriate feature representations integration methods can result in information repetition and redundancy. It will have a negative impact on integrating information. Therefore, we evaluate the performance of different designing merge methods: addition, connection, unit-level attention, and channel-level attention. From Table 3, when the unit-level attention method is adopted, the model obtains the highest $F1$-Score. The probable reason is that the unit-level attention method can fully integrate the important features captured by each channel and do not interfere with each other; thus, we use the unit-level attention method in the merge layer.

*4.2. Performance Comparison of each Representations.* This paper proposes a multichannel BiGRU-CRF model to capture richer feature information by sending multiple representations individually or collectively into BiGRU. We evaluate the performance of each channel based on different representations while verifying the effectiveness of our multichannel method. The experimental results are shown in Table 4. It can be seen that the multichannel representations can provide richer potential information, and the concatenate representations are superior to the single representations. In summary, we compare the performance between each representation on the same corpus. Our merge-based multiple representations method achieves optimal performance, with the F1-scores of 76.0 and 88.7 on the JNLPBA and NCBI-disease corpora, respectively.

*4.3. Performance Comparison of GRU Units Dimensions.* GRU units' dimensions affect the ability of learning features and the performance of the classifier. Too few hidden units can result in insufficient capture features. Conversely, it may lead to information redundancy and increase the

Table 2: Introduction to experimental corpora.

| Main | Entity types and counts | Size |
|---|---|---|
| NCBI-disease | Disease (6881) | 793 |
| JNLPBA | Gene/proteins (35336); cell line (4330); cell type (8649); DNA(10589); RNA(1069) | 2404 |
| **Auxiliary** | **Entity types and counts** | **Size** |
| BC5CDR-disease | Disease (12852) | 1500 |
| BC2GM | Gene/proteins (24583) | 20000 |

Table 3: Performance comparison of the different merge methods.

| Merge methods | JNLPBA | | | NCBI-disease | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Addition | 72.9 | 78.8 | 75.7 | 87.4 | 88.6 | 88.0 |
| Connection | 71.1 | 78.3 | 74.5 | 85.3 | 89.2 | 87.2 |
| **Unit-level attention** | **72.6** | **79.6** | **76.0** | **88.2** | **89.2** | **88.7** |
| Channel-level attention | 72.1 | 78.6 | 75.2 | 86.6 | 88.7 | 87.6 |

Table 4: Performance comparison of each representations.

| JNLPBA | Precision | Recall | F1-score | Δ | NCBI-disease | Precision | Recall | F1-score | Δ |
|---|---|---|---|---|---|---|---|---|---|
| **Ours** | **72.6** | **79.6** | **76.0** | — | **Ours** | **88.2** | **89.2** | **88.7** | — |
| ELMo | 69.8 | 76.8 | 73.1 | 2.9 | ELMo | 83.8 | 85.4 | 84.6 | 4.1 |
| Char | 66.9 | 71.9 | 69.3 | 6.7 | Char | 83.7 | 80.6 | 82.1 | 6.6 |
| Word | 69.9 | 75.5 | 72.6 | 3.4 | Word | 84.2 | 80.9 | 82.5 | 6.2 |
| ELMo+Char | 71.5 | 76.4 | 73.8 | 2.2 | ELMo+Char | 86.0 | 85.5 | 85.7 | 3.0 |
| Word+Char | 68.9 | 77.6 | 73.0 | 3.0 | Word+Char | 84.2 | 85.1 | 84.7 | 4.0 |
| Word+ELMo | 70.1 | 77.5 | 73.6 | 2.4 | Word+ELMo | 84.5 | 86.0 | 85.3 | 3.4 |
| Word+ELMo+Char | 71.4 | 77.8 | 74.4 | 1.6 | Word+ELMo+Char | 87.2 | 85.9 | 86.6 | 2.1 |

Table 5: Performance comparison of GRU units' dimensions.

| GRU | JNLPBA | Precision | Recall | F1-score | NCBI-disease | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|---|---|
| Dimensions | 50 | 70.7 | 77.5 | 73.9 | 50 | 87.0 | 87.5 | 87.2 |
| | **100** | 72.6 | 79.6 | 76.0 | **100** | 88.2 | 89.2 | 88.7 |
| | 150 | 70.3 | 77.3 | 73.6 | 150 | 88.1 | 85.7 | 86.9 |
| | 200 | 71.1 | 76.3 | 73.6 | 200 | 85.1 | 86.9 | 86.0 |

Table 6: Performance comparison of different optimization methods.

| | Precision | Recall | F1-score |
|---|---|---|---|
| JNLPBA | | | |
| SGD | 64.9 | 73.7 | 68.9 |
| AdaGrad | 72.0 | 75.0 | 73.4 |
| Adam | 72.6 | 79.6 | 76.0 |
| NCBI-disease | | | |
| SGD | 77.5 | 79.8 | 78.6 |
| AdaGrad | 85.4 | 86.5 | 85.9 |
| Adam | 88.2 | 89.2 | 88.7 |

computational burden. Both of them will have a negative impact on model performance. Therefore, we evaluate the performance of different neuron dimensions to obtain the best hyperparameters. We set the size of GRU units to be 50, 100, 150, 200 and evaluate them. As the results show in Table 5, when the dimensions are 100, it achieves the best performance. Therefore, the GRU units' dimensions are set to 100.

*4.4. Performance Comparison of Combining CRF Layer.* The CRF layer can capture the dependence between adjacent labels by transition probability. This paper evaluates the effectiveness of the CRF layer. The experimental results are shown in Table 8. After combining BiGRU with the CRF layer, the model performance has been significantly

Table 7: Performance comparison of using dropout.

| | Precision | Recall | F1-score |
|---|---|---|---|
| JNLPBA | | | |
| No | 73.5 | 72.9 | 73.2 |
| Yes | 72.6 | 79.6 | 76.0 |
| Δ | **-0.9** | **+6.7** | **+2.8** |
| NCBI-disease | | | |
| No | 86.9 | 84.8 | 85.9 |
| Yes | 88.2 | 89.2 | 88.7 |
| Δ | **+1.3** | **+4.4** | **+2.8** |

Table 8: Performance comparison of model with and without CRF layer.

| | Precision | Recall | F1-score |
|---|---|---|---|
| JNLPBA | | | |
| BiGRU | 70.8 | 73.7 | 72.3 |
| BiGRU-CRF | 72.6 | 79.6 | 76.0 |
| Δ | **+1.8** | **+5.9** | **+3.7** |
| NCBI-disease | | | |
| BiGRU | 82.4 | 86.7 | 84.5 |
| BiGRU-CRF | 88.2 | 89.2 | 88.7 |
| Δ | **+5.8** | **+2.5** | **+4.2** |

Table 9: Performance comparison of adopting multitask learning.

| | Precision | Recall | F1-score |
|---|---|---|---|
| JNLPBA | | | |
| Single-task | 71.8 | 79.7 | 75.6 |
| Multi-task | 72.6 | 79.6 | 76.0 |
| Δ | **+0.8** | **-0.1** | **+0.4** |
| NCBI-disease | | | |
| Single-task | 87.2 | 88.6 | 87.9 |
| Multi-task | 88.2 | 89.2 | 88.7 |
| Δ | **+1.0** | **+0.6** | **+0.8** |

improved on the JNLPBA and NCBI-disease corpora. It proves the validity of the CRF layer.

*4.5. Performance Comparison of Adopting Multitask Learning.* From the Table 9, the multitask learning strategy we adopted is effective. The auxiliary corpora provide more training samples and valuable information for the main corpora. According to the analysis of main corpora evaluation results, the multitask learning framework makes the performance improvement of JNLPBA less obvious than NCBI-disease. The possible reason is that the entity type of NCBI-disease is completely consistent with the auxiliary corpus BC5CDR-disease. The auxiliary corpus BC2GM contains only "protein", the training samples and relevant information of the other four entity types in the main corpus JNLPBA have not been supplemented.

*4.6. Performance Comparison of Optimization Methods.* The optimization method determines the convergence speed and performance of the model training process. This paper evaluates three different optimization methods: Adam, SGD, and AdaGrad. SGD is one of the commonly used optimizers during training. It randomly extracts fixed-size training samples to calculate gradients and update parameters. But it may lead to convergence to a local minimum. Compared to SGD, AdaGrad does not rely on a preset learning rate, but adaptively adjusts it during training. It is well suited to handle sparse data but may cause a vanishing gradient. The experimental results are shown in Table 6. Compared with the other two optimization methods, Adam achieves the fastest convergence speed and highest performance under the same conditions. Therefore, this paper uses Adam as the optimizer.

*4.7. Performance Comparison of Using Dropout.* This paper evaluates the effectiveness of dropout. The experimental results are shown in Table 7. After setting the dropout rate, the model performance has been significantly improved on the JNLPBA and NCBI-disease corpora. It demonstrates the validity of dropout.

*4.8. Performance Comparison with Existing Feature-Based Methods.* Lastly, we draw a comparison between our model and existing models. In order to ensure the fairness and rationality of the experiment, we have divided the existing models into two kinds according to the different training patterns. One kind is feature-based, which applies specific input representations to task-specific different architectures, such as the approaches listed in Table 10; while another kind is fine-tuning, which trains various downstream tasks with fine-tuning parameters in fixed model architectures, such as BERT [55]. This paper reports the performance comparison with existing models of feature-based representations.

The performance comparison results on the JNLPBA corpus are shown on the left side of Table 10. In these studies, the early methods (dictionary based and rule based) and the conventional machine learning models also obtained reasonable results in BNER, including Finkel et al. [1], Settles [3], Tsuruoka et al. [2], Tang et al. [6], Chang et al. [4], and Liao et al. [5]. NERBio [53] was the best rule-based system on a JNLPBA corpus, and the F1-score is 73.0. The Skip-Chain CRF adopted by Liao et al. [5] was the state-of-the-art conventional machine learning model. It obtained a reasonable F1-score of 73.2. Compared with the above best early method and conventional machine learning method, our model has increased F1-score values by 3.0 and 2.8, respectively. We can produce these results without any feature engineering but simple architecture. Compared with existing deep learning studies, the performance of our model is better than Li et al. [33]. They proposed a CNN-BLSTM-CRF model with word embeddings and character embeddings. Our model has increased the recall and F1-score by 9.7 and 1.6, respectively. Gridach et al. [54] proposed a BiLSTM-CRF model

Table 10: Performance comparison with existing feature-based methods.

| Methods | JNLPBA | | | | Methods | NCBI-disease | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Type | *P* | *R* | *F*1 | | Type | *P* | *R* | *F*1 |
| Finkel et al. [1] | S | 71.6 | 68.6 | 70.1 | Xu et al. [48] | S | 84.8 | 76.1 | 80.2 |
| Settles [3] | S | 69.1 | 72.0 | 70.5 | Leaman et al. [49] | S | 82.8 | 81.9 | 80.9 |
| Yao et al. [18] | S | 64.9 | 76.1 | 71.0 | Dogan et al. [45] | S | 83.8 | 80.0 | 81.8 |
| Tsuruoka et al. [2] | S | 67.5 | 75.8 | 71.4 | Leaman et al. [50] | S | 85.1 | 80.8 | 82.9 |
| Tang et al. [6] | S | 70.8 | 72.0 | 71.4 | Limsopatham et al. [22] | S | 86.7 | 81.9 | 84.3 |
| Chang et al. [4] | S | — | — | 71.9 | Wei et al. [51] | S | 85.3 | 83.3 | 84.3 |
| Zhu et al. [20] | S | — | — | 72.6 | Dang et al. [24] | S | 85.0 | 83.8 | 84.4 |
| Li et al. [21] | S | 74.8 | 70.9 | 72.8 | Habibi et al. [52] | S | 86.4 | 82.9 | 84.6 |
| Tsai et al. [53] | S | 72.0 | 74.0 | 73.0 | Zhao et al. [19] | S | 85.1 | 85.3 | 85.2 |
| Liao et al. [5] | S | 72.8 | 73.6 | 73.2 | Wang et al. [26] | M | 85.9 | 86.4 | 86.1 |
| Wang et al. [26] | M | 70.9 | 76.3 | 73.5 | Xu et al. [23] | S | 86.6 | 85.8 | 86.2 |
| Lyu et al. [25] | S | 71.2 | 76.5 | 73.8 | Yoon et al. [27] | M | 85.5 | 87.3 | 86.4 |
| Li et al. [33] | S | 79.6 | 69.9 | 74.4 | Zhu et al. [20] | S | 86.5 | 88.1 | 87.3 |
| Gridach et al. [54] | S | 74.1 | 77.7 | 75.8 | Sachan et al. [28] | T | 86.4 | 88.3 | 87.3 |
| **Ours** | **M** | **72.6** | **79.6** | **76.0** | **Ours** | **M** | **88.2** | **89.2** | **88.7** |

∗"S" denotes the single-task model. "M" denotes the multitask model. "T" denotes the model based on transfer learning.

with pretrained word embeddings and character embeddings. They computed the character vectors by a bidirectional LSTM. This model significantly enhanced the best performance of single-task BNER models. The performance of our model is close to theirs. In summary, our method obtains promising results compared with existing feature-based models under the premise of using merge-based multiple features and simple architecture.

The performance comparison on the NCBI-disease corpus is shown in Table 10 (right side). In these studies, Leaman et al. [45, 49, 50] first adopted conventional machine learning methods to obtain competitive performance on the NCBI-disease dataset. They developed multiple BNER systems (e.g., DNorm and TaggerOne) in subsequent studies. The recent deep learning methods achieved satisfactory results in BNER. In addition to some of the related works described in the first section, including Limsopatham et al. [22], Dang et al. [24], Zhao et al. [19], Wang et al. [26], Xu et al. [23], Yoon et al. [27], Zhu et al. [20], and Sachan et al. [28], Xu et al. [48] proposed a three-layer neural network to identify disease entities. The BiLSTM with the same structure was used to generate character-level embeddings and capturing features. The entity labels were predicted through the CRF layer. Wei et al. [51] designed a hybrid model combining the conventional machine learning methods with neural networks, and bidirectional RNN and CRF were employed as submodels to extract features. Then, the output was merged and fed into SVM for classification. Habibi et al. [52] achieved reasonable performance on multiple biomedical datasets based on word embedding and a LSTM-CRF model. GRAM-CNN [20] was the best single-task system which was developed by CNN on the NCBI-disease corpus. It obtained an F1-score of 87.3. BiLM-NER [28] was the best feature-based model and was developed by the transfer learning method; the F1-score was 87.3. However, our model's performance is better than the above state-of-the-art work. Our model obtains the best performance among reported existing feature-based models.

*4.9. Error Analysis.* We analyze the error cases of the model on our corpora and summarized the main causes of these errors into the following two points.

The boundary is blurred. There are 3 main reasons for this error. First, biomedical entities are generally long and complex. For example, "Kappa B-specific DNA binding proteins" contains five words as the entity, and the length of entities in the general field is usually within three words. In addition, it contains the word "DNA", and the entity itself is "protein". Second, the virtual words and conjunctions within biomedical entities influence the judgment of the boundary. For example, there may be fixed-use conjunctions in biomedical entities, but they are often misjudged as "O". Finally, an entity in biomedical corpora is part of another entity, but they belong to two types. For example, "MZF-1" is part of "Recombinant MZF-1", but they belong to "DNA" and "protein". To a certain extent, these above issues are plaguing our model.

Corpora annotation inconsistency. For example, "wild-type" is labeled as "O" in "gave nearly wild-type levels of gene expression in phorbol ester-treated Jurkat cells but not in phorbol ester-treated HeLa or U937 cells.", but in "as a wild-type but not a mutant TSAP-binding site of the sea urchin functions only in transfected B cells as an upstream promoter element.", it is labeled as "DNA". In addition, there are abbreviations of entities in some biomedical sequences, and our model is difficult to identify. For example, "IL-2" in "Under the same conditions, Lck did not stimulate IL-2 promoter unless it was activated by mutation" and "Interleukin-2" in "The proteasome regulates receptor-mediated endocytosis of interleukin-2" refer to the same entity, but our model has difficulty to distinguish them.

These analyses demonstrate that the complexity and annotation inconsistency of biomedical corpora are major

factors that result in errors. To address these issues, we can disambiguate through entity linking during corpora preprocessing or adopt more external representations.

## 5. Conclusion

In this paper, we propose a multitask learning multichannel BiGRU-CRF model based on contextualized word representations. First, we obtain word, character, and contextualized word representations through a biomedical pretrained dictionary, convolutional neural networks, and ELMo pretrained language model, respectively. The character representations can describe the morphological features of words, and the contextualized word representations can better represent both polysemous and ambiguous words according to the context information. Then, we train multiple BiGRU submodels at the same time, each of which is viewed as a channel. The three representations are used as input for different channels, respectively, or in combination. Next, we design multiple methods to integrate the output of each channel in the merge layer. Finally, considering the dependence between labels, the CRF layer is adopted to parse sequence labels. It avoids outputting non-compliant label sequences. In addition, multitask learning strategy is adopted to solve the problem of insufficient training samples in specific fields. The auxiliary corpora with the same entity types are applied to supplement more training samples and relevant information for the main corpora to be evaluated. Our model has a simple architecture and avoids feature engineering. The multitask learning multichannel BiGRU-CRF achieves promising results on JNLPBA and NCBI-disease corpora, with F1-scores of 76.0 and 88.7, respectively. In the future, we plan to introduce more abundant additional features (e.g., domain knowledge base, structured ontology) to enhance the performance.

## Data Availability

The data sets used in this paper are all publicly available. The related references of data sets adopted to support the findings of this study are included within this paper.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.61976124).

## References

[1] J. Finkel, S. Dingare, H. Nguyen, M. Nissim, C. Manning, and G. Sinclair, "Exploiting context for biomedical entity recognition: From syntax to the web," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pp. 88–91, Geneva, Switzerland, 2004, Association for Computational Linguistics.

[2] Y. Tsuruoka, Y. Tateishi, J. D. Kim et al., "Developing a robust part-of-speech tagger for biomedical text," in *Panhellenic Conference on Informatics*, pp. 382–392, Volas, Greece, 2005, Springer.

[3] B. Settles, "Abner: an open source tool for automatically tagging genes, proteins and other entity names in text," *Bioinformatics*, vol. 21, no. 14, pp. 3191-3192, 2005.

[4] F. Chang, J. Guo, W. Xu, and S. R. Chung, "Application of word embeddings in biomedical named entity recognition tasks," *Journal of Digital Information Management*, vol. 13, no. 5, 2015.

[5] Z. Liao and H. Wu, "Biomedical named entity recognition based on skip-chain crfs," in *2012 International Conference on Industrial Control and Electronics Engineering*, pp. 1495–1498, Xi'an, China, 2012.

[6] B. Tang, H. Cao, X. Wang, Q. Chen, and H. Xu, "Evaluating word representation features in biomedical named entity recognition tasks," *BioMed Research International*, vol. 2014, Article ID 240403, 6 pages, 2014.

[7] K. Li, W. Ai, Z. Tang et al., "Hadoop recognition of biomedical named entity using conditional random fields," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 11, pp. 3040–3051, 2015.

[8] M. Chen and Y. Hao, "Label-less learning for emotion cognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, pp. 1–11, 2019.

[9] M. Chen, Y. Hao, H. Gharavi, and V. C. M. Leung, "Cognitive information measurements: a new perspective," *Information Sciences*, vol. 505, pp. 487–497, 2019.

[10] M. Chen, Y. Jiang, Y. Cao, and A. Y. Zomaya, "Creativebioman: Brain and body wearable computing based creative gaming system," 2019, https://arxiv.org/abs/1906.01801.

[11] M. Chen, Y. Jiang, N. Guizani et al., "Living with i-fabric: smart living powered by intelligent fabric and deep analytics," *IEEE Network*, pp. 1–8, 2020.

[12] Y. Chen, J. Tao, Q. Zhang et al., "Saliency detection via the improved hierarchical principal component analysis method," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 8822777, 12 pages, 2020.

[13] Y. Chen, J. Wang, S. Liu et al., "Multiscale Fast Correlation Filtering Tracking Algorithm Based on a Feature Fusion Model," *Concurrency and Computation: Practice and Experience*, 2019.

[14] Y. Chen, W. Xu, J. Zuo, and K. Yang, "The fire recognition algorithm using dynamic feature fusion and iv-svm classifier," *Cluster Computing*, vol. 22, pp. 7665–7675, 2019.

[15] Y. Zhang, X. Ma, J. Zhang, M. S. Hossain, G. Muhammad, and S. U. Amin, "Edge intelligence in the cognitive internet of things: improving sensitivity and interactivity," *IEEE Network*, vol. 33, no. 3, pp. 58–64, 2019.

[16] Y. Zhang, Y. Qian, D. Wu, M. S. Hossain, A. Ghoneim, and M. Chen, "Emotion-aware multimedia systems security," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 617–624, 2019.

[17] Y. Zhang, R. Wang, M. S. Hossain, M. F. Alhamid, and M. Guizani, "Heterogeneous information network-based content caching in the internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10216–10226, 2019.

[18] L. Yao, H. Liu, Y. Liu, X. Li, and M. W. Anwar, "Biomedical named entity recognition based on deep neutral network," *International Journal of Hybrid Information Technology*, vol. 8, no. 8, pp. 279–288, 2015.

[19] Z. Zhao, Z. Yang, L. Luo et al., "Disease named entity recognition from biomedical literature using a novel convolutional neural network," *BMC Medical Genomics*, vol. 10, no. S5, 2017.

[20] Q. Zhu, X. Li, A. Conesa, and C. Pereira, "GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text," *Bioinformatics*, vol. 34, no. 9, pp. 1547–1554, 2018.

[21] L. Li, L. Jin, Y. Jiang, and D. Huang, "Recognizing biomedical named entities based on the sentence vector/twin word embeddings conditioned bidirectional lstm," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pp. 165–176, Springer, 2016.

[22] N. Limsopatham and N. Collier, "Learning orthographic features in bi-directional lstm for biomedical named entity recognition," in *Proceedings of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM 2016)*, pp. 10–19, Osaka, Japan, 2016.

[23] K. Xu, Z. Zhou, T. Gong, T. Hao, and W. Liu, "SBLC: a hybrid model for disease named entity recognition based on semantic bidirectional LSTMs and conditional random fields," *BMC Medical Informatics and Decision Making*, vol. 18, no. S5, 2018.

[24] T. H. Dang, H. Q. Le, T. M. Nguyen, and S. T. Vu, "$D_3$ner: biomedical named entity recognition using crf-bilstm improved with fine-tuned embeddings of various linguistic information," *Bioinformatics*, vol. 34, no. 20, pp. 3539–3546, 2018.

[25] C. Lyu, B. Chen, Y. Ren, and D. Ji, "Long short-term memory RNN for biomedical named entity recognition," *BMC Bioinformatics*, vol. 18, no. 1, 2017.

[26] X. Wang, Y. Zhang, X. Ren et al., "Cross-type biomedical named entity recognition with deep multi-task learning," 2018, https://arxiv.org/abs/1801.09851.

[27] W. Yoon, C. H. So, J. Lee, and J. Kang, "Collabonet: collaboration of deep neural networks for biomedical named entity recognition," 2018, https://arxiv.org/abs/1809.07950.

[28] D. S. Sachan, P. Xie, M. Sachan, and E. P. Xing, "Effective use of bidirectional language modeling for transfer learning in biomedical named entity recognition," 2017, https://arxiv.org/abs/1711.07908.

[29] M. E. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," 2018, https://arxiv.org/abs/1802.05365.

[30] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[31] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP*, pp. 1532–1543, Doha, Qatar, 2014.

[32] C. Quan, L. Hua, X. Sun, and W. Bai, "Multichannel convolutional neural network for biological relation extraction," *BioMed Research International*, vol. 2016, Article ID 1850404, 10 pages, 2016.

[33] L. Li and Y. Guo, "Biomedical named entity recognition with cnn-blstm-crf," *Journal of Chinese Information Processing*, vol. 32, no. 1, pp. 116–122, 2018.

[34] D. Zeng, C. Sun, L. Lin, and B. Liu, "Lstm-crf for drug-named entity recognition," *Entropy*, vol. 19, no. 6, 2017.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, https://arxiv.org/abs/1409.1259.

[37] K. Cho, B. Van Merriënboer, C. Gulcehre et al., "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[38] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.

[39] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[40] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," in *International conference on machine learning*, pp. 2342–2350, Lille, France, 2015.

[41] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," 2016, https://arxiv.org/abs/1603.01360.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[43] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[44] J. D. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier, "Introduction to the bio-entity recognition task at JNLPBA," in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications - JNLPBA '04*, pp. 70–75, Geneva, Switzerland, 2004.

[45] R. I. Doğan, R. Leaman, and Z. Lu, "Ncbi disease corpus: a resource for disease name recognition and concept normalization," *Journal of Biomedical Informatics*, vol. 47, pp. 1–10, 2014.

[46] L. Smith, L. K. Tanabe, R. J. n. Ando et al., "Overview of biocreative ii gene mention recognition," *Genome Biology*, vol. 9, no. S2, 2008.

[47] C. H. Wei, Y. Peng, R. Leaman et al., "Overview of the biocreative v chemical disease relation (cdr) task," in *Proceedings of the fifth BioCreative challenge evaluation workshop*, vol. 14, Seville, Spain, 2015.

[48] K. Xu, Z. Zhou, T. Hao, and W. Liu, "A bidirectional lstm and conditional random fields approach to medical named entity recognition," in *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017*, pp. 355–365, Cairo, Egypt, 2018, Springer.

[49] R. Leaman, R. I. Dogan, and Z. Lu, "Dnorm: disease name normalization with pairwise learning to rank," *Bioinformatics*, vol. 29, no. 22, pp. 2909–2917, 2013.

[50] R. Leaman and Z. Lu, "Taggerone: joint named entity recognition and normalization with semi-markov models," *Bioinformatics*, vol. 32, no. 18, pp. 2839–2846, 2016.

[51] Q. Wei, T. Chen, R. Xu, Y. He, and L. Gui, "Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks," *Database*, vol. 2016, article baw140, 2016.

[52] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, "Deep learning with word embeddings improves biomedical named entity recognition," *Bioinformatics*, vol. 33, no. 14, pp. i37–i48, 2017.

[53] R. Tsai, C.-L. Sung, H.-J. Dai, H.-C. Hung, T.-Y. Sung, and W.-L. Hsu, "NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition," *BMC Bioinformatics*, vol. 7, Supplementary 5, 2006.

[54] M. Gridach, "Character-level neural network for biomedical named entity recognition," *Journal of Biomedical Informatics*, vol. 70, pp. 85–91, 2017.

[55] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018, https://arxiv.org/abs/1810.04805.