Original article

Chemical–protein interaction extraction via contextualized word representations and multihead attention

Yijia Zhang ID ***, Hongfei Lin, Zhihao Yang, Jian Wang and Yuanyuan Sun**

College of Computer Science and Technology, Dalian University of Technology, Dalian 10064, China

*Corresponding author: Tel.: +86-0411-84706550; Fax: +86-0411-84709144; Email: zhyj@dlut.edu.cn

Abstract

A rich source of chemical–protein interactions (CPIs) is locked in the exponentially growing biomedical literature. Automatic extraction of CPIs is a crucial task in biomedical natural language processing (NLP), which has great benefits for pharmacological and clinical research. Deep context representation and multihead attention are recent developments in deep learning and have shown their potential in some NLP tasks. Unlike traditional word embedding, deep context representation has the ability to generate comprehensive sentence representation based on the sentence context. The multihead attention mechanism can effectively learn the important features from different heads and emphasize the relatively important features. Integrating deep context representation and multihead attention with a neural network-based model may improve CPI extraction. We present a deep neural model for CPI extraction based on deep context representation and multihead attention. Our model mainly consists of the following three parts: a deep context representation layer, a bidirectional long short-term memory networks (Bi-LSTMs) layer and a multihead attention layer. The deep context representation is employed to provide more comprehensive feature input for Bi-LSTMs. The multihead attention can effectively emphasize the important part of the Bi-LSTMs output. We evaluated our method on the public ChemProt corpus. These experimental results show that both deep context representation and multihead attention are helpful in CPI extraction. Our method can compete with other state-of-the-art methods on ChemProt corpus.

Database URL: https://github.com/zhangyijia1979/CPI_extraction

Introduction

Accurately detecting the interactions between chemicals and proteins is a crucial task that plays a key role in precision medicine, drug discovery and basic clinical research (1). Currently, PubMed contains >28 million articles, and its annual growth rate is more than a million articles each year. A large amount of valuable chemical–protein interactions (CPIs) are hidden in the biomedical literature. There is an increasing interest in CPI extraction from the biomedical literature.

Since manually extracting biomedical relations such as protein–protein interactions (PPI) and drug–drug interactions (DDI) is costly and time-consuming, some computational methods (2–6) have been successfully proposed for automatic biomedical relation extraction. For example, Kim *et al.* (4) proposed using a subsequence kernel for PPI extraction that matches the e-walk and v-walk on the shortest dependency to capture the noncontiguous syntactic structures. Segura-Bedmar *et al.* (7) employed linguistic patterns to extract DDIs. Currently, models based on deep neural networks have exhibited surprising potential in biomedical relation extraction (8–10). Rois *et al.* (11) proposed an adversarial domain adaptation method to extract PPIs and DDIs. Zhang *et al.* (12) proposed a hybrid deep neural model for biomedical relation extraction from the biomedical literature, which integrates the advantages of convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

To date, most studies on the biomedical relation extraction have focused on the PPIs and DDIs, but a few attempts have been made to extract CPIs. The BioCreative VI ChemProt shared task (13) released the ChemProt dataset for CPI extraction, which is the first challenge for extracting CPIs. The ChemProt dataset (13) provided an opportunity to compare current CPI extraction methods on the same benchmark corpora. Peng *et al.* (14) proposed an ensemble method to integrate the support vector machines (SVMs) and deep neural networks and achieved an *F*-score of 0.641 on the ChemProt dataset. Corbett and Boyle (15) employed transfer learning and specialized word embeddings to extract CPIs and achieved an *F*-score of 0.615 on the ChemProt dataset.

From the BioCreative VI ChemProt track, neural network-based methods achieved state-of-the-art performance in CPI extraction. Compared with feature-based and kernel-based methods, the deep neural networks can automatically learn latent features. So far, the best performance of CPI extraction is an *F*-score of 0.641 on the ChemProt corpus (14). One of the bottlenecks of neural networks in natural language processing (NLP) is word embeddings that are generally the input layer of various neural networks. Pretrained word embeddings (16, 17) are of great importance for the performance of neural network-based methods in NLP tasks. Learning high-quality distributed word embeddings is very challenging. Although great efforts have been made in distributed word representations, current word embeddings still cannot effectively vary across linguistic contexts. Most recently, Peters *et al.* (18) proposed deep contextualized word representations called ELMo based on a deep bidirectional language model. Traditional word embeddings represent each token as a unique embedding vector. However, ELMo represents each token as a function of the entire input sentence, which makes the representation of each token dependent on the sentence context. Therefore, integrating the ELMo representation with deep neural networks can provide more comprehensive input representation for the following neural network models and may improve the performance of CPI extraction.

Another challenge in CPI extraction is how to accurately detect and extract the CPIs in long and complicated sentences. In particular, the chemical and protein entities are often found in different clauses. It is hard to capture the distinguished syntactic information for deep neural networks in these long and complicated sentences. Recent studies (19, 20) have suggested attention mechanisms can effectively emphasize the relatively important parts of the input sentences and be helpful in boosting the performance of relation extraction. However, most studies only employed single attention in the deep neural models. Multihead attention applies attention multiple times and divides attention information into multiple heads (21). Thus, a multihead attention mechanism will make it easier to capture the relevant important information for deep neural networks in CPI extraction.

In this work, we explore the effectiveness of deep contextualized word representations and multihead self-attention mechanisms in the CPI extraction. We introduce a deep neural model to extract CPIs from the literature, which includes an ELMo input layer, bidirectional long short-term memory networks (Bi-LSTMs) and a multihead attention layer. Liu *et al.* (22) integrated attention pooling into the gated recurrent unit (GRU) model to extract CPIs. Verga *et al.* (23) combined the multihead attention with CNNs to construct transformer model to extract the document-level biomedical relations. In this work, we combined the multihead attention with Bi-LSTMs. In particular, we employed the ELMo contextualized representation in the input layer. To the best of our knowledge, this is the first model that used ELMo contextualized representation for biomedical relation extraction. Our proposed model is evaluated on the ChemProt corpus. The experimental results show that both contextualized word representations and multihead atten-

tion are valuable for CPI extraction. Our model can effectively integrate the contextualized word representations and multihead attention for CPI extraction and achieve state-of-the-art performance on ChemProt corpus. We have also shown that our model can also achieve competitive performance on other biomedical relation extraction tasks such as DDI extraction.

Materials and methods

CPI extraction

Computational CPI extraction is generally approached as a task of classifying whether a specified semantic relation holds between the chemical and protein entities within a sentence or document. The ChemProt corpus is a manually annotated CPI dataset, which greatly promotes the development of CPI extraction approaches. The ChemProt corpus includes training, development and test sets. Each set contains the corpus file, entity annotation file and relation annotation file. Figure 1 gives an example of the corpus file, entity annotation file and relation annotation file from the ChemProt training set. The corpus file gives the abstract document and the PubMed Unique Identifier (PMID) (10480573). The entity file gives all chemical and gene/protein entity mentions in the abstract. The entity annotation information includes the PMID, entity number, type of entity mention ('CHEMICAL', 'GENE-Y' and 'GENE-N'), start and end

Abstract example

10480573   Effect of thiazinotrienomycin B, an ansamycin antibiotic, on the function of epidermal growth factor receptor in human stomach tumor cells. Thiazinotrienomycin B (TT-B), an ansamycin isolated from fermentation broths of Streptomyces sp. MJ672-m3, inhibited the growth in vitro of human stomach tumor SC-6 cells over 10 times more strongly than the growth of other human tumor cells, such as HeLa (cervix), T24 (bladder) and LX-1 (lung). The extent of growth inhibition by TT-B of SC-6, but not of LX-1 nor T24, was lowered in a competitive manner by raising serum concentrations in the culture medium. TT-B inhibited the cell cycle progression of SC-6 at an early stage of the progression from G0/G1 to S. The inhibition was again competitive with serum concentrations in the culture medium. No direct inhibition of DNA synthesis was observed at the concentration range which caused the cell cycle arrest. TT-B and anti-epidermal growth factor receptor (anti-EGFR) were antagonistic to each other in inhibiting the cell cycle progression of SC-6 from G0/G1 to S, suggesting that the two compounds share the same target, EGFR. The kinase activity of EGFR was little inhibited by TT-B in a cell-free system.

Entity annotation example

10480573	T1	CHEMICAL	140	161	Thiazinotrienomycin B
10480573	T2	CHEMICAL	1178	1182	TT-B
10480573	T3	CHEMICAL	163	167	TT-B
10480573	T4	CHEMICAL	173	182	ansamycin
10480573	T5	CHEMICAL	602	606	TT-B
10480573	T6	CHEMICAL	906	910	TT-B
10480573	T7	CHEMICAL	10	31	thiazinotrienomycin B
10480573	T8	CHEMICAL	36	45	ansamycin
10480573	T9	GENE-Y	1149	1153	EGFR
10480573	T10	GENE-Y	920	952	epidermal growth factor receptor
10480573	T11	GENE-Y	959	963	EGFR
10480573	T12	GENE-Y	1120	1124	EGFR
10480573	T13	GENE-N	1130	1136	kinase
10480573	T14	GENE-Y	77	109	epidermal growth factor receptor

CPI annotation example

| 10480573 | CPR:4 | Y | INHIBITOR | Arg1:T2 | Arg2:T13 |
| 10480573 | CPR:4 | Y | INHIBITOR | Arg1:T2 | Arg2:T9 |

Figure 1. The illustrative examples of ChemProt corpus.

Table 1. The ChemProt corpus semantic relations

Relation class	Eval.	ChemProt relations
CPR:1	N	Part of
CPR:2	N	Regulator
CPR:3	Y	Upregulator and activator
CPR:4	Y	Downregulator and inhibitor
CPR:5	Y	Agonist
CPR:6	Y	Antagonist
CPR:7	N	Modulator
CPR:8	N	Cofactor
CPR:9	Y	Substrate and product of
CPR:10	N	Not

character offset of the entity mention and text string of entity mention. 'GENE-Y' denotes the entity mention can be associated with a biological database identifier. 'GENE-N' denotes the entity mention cannot be associated with a biological database identifier. The relation file gives the detailed CPI annotations in the abstract. It contains PMID, CPI relation class, evaluation type (Y: evaluated; N: not evaluated), CPI relation and interactor arguments. In Figure 1, it can be seen that 14 chemical or gene/protein entities and 2 CPIs are annotated in the abstract (PMID: 10480573). In the BioCreative VI ChemProt share task, the named entity recognition has been already done, and the participant teams only focused on CPI extraction.

Table 1 shows the 10-type relation classes of the ChemProt corpus. In Table 1, it can be seen that each relation class includes one or multiple relation types. 'Eval.' denotes whether the relation class is evaluated in the BioCreative VI ChemProt share task. Although the ChemProt corpus contains 10-type relation classes, only 5-type relation classes were evaluated, including CPR:3, CPR:4, CPR:5, CPR:6 and CPR:9. Table 2 shows the statistics of the ChemProt corpus.

Figure 2 gives some illustrative examples of CPI extraction. In example 1, 'Ibandronate' and 'FAS' (presented in bold) are chemical and gene entities, respectively. To accurately extract the CPI from example 1, we need not only to detect the interaction between 'Ibandronate' and 'FAS' but to also classify the interaction as the 'CPR:3' class. There are some long and complicated sentences in

ChemProt corpus. Example 6 is a long sentence instance that contains multiple subsentences.

The model architecture

Figure 3 is a schematic overview of our model. In a nutshell, our model mainly includes three parts: the deep contextualized representation layer, the Bi-LSTMs layer and the multihead attention layer. The inputs of our model are sentence sequences. The deep contextualized representation layer will generate the contextualized representation vector for each word based on the sentence context. Some recent studies (8, 24) have suggested that the position and part of speech (POS) of each word in the sentence are crucial to biomedical relation extraction. Hence, the word contextualized representation is concatenated with position and POS embeddings. The Bi-LSTMs layer will learn the latent features based on the whole word representations. The multihead attention layer applies a self-attention mechanism to capture the relative important features in the CPI extraction. After the multihead attention layer, we employed attention pooling and a softmax function to implement the detection and classification of the candidate CPIs in the sentences. In the following section, our CPI extraction model will be described in detail.

Contextualized word representations

In our model, we used deep contextualized word (ELMo) representations instead of the traditional word embeddings. Unlike word embeddings, ELMo representations are functions of the input sentences based on bidirectional language models. Therefore, ELMo has the ability to generate different representation vectors for the same word according to the sentence context.

Given a sentence S, $\{w_1, w_2, \ldots, w_n\}$ denotes the word sequence in the sentence. Given a word w_k, the forward language model calculates the probability of the word w_k based on the front words $\{w_1, w_2, \ldots, w_{k-1}\}$ of w_k in S as follows:

$$P_{forward}(w_1, w_2, \ldots, w_n) = \prod_{k=1}^{n} p(w_k | w_1, w_2, \ldots, w_{k-1}). \quad (1)$$

Table 2. The statistics of the ChemProt corpus

Dataset	Abstracts	Chemicals	Genes/proteins	Evaluated CPIs
Training	1020	13 017	12 735	4157
Development	612	8004	7563	2416
Test	3399	10 810	10 018	3458
Total	5031	31 813	30 316	10 031

CPR:3 (Up regulator/Activator)

Example 1: **Ibandronate** increases the expression of the pro-apoptotic gene **FAS** by epigenetic mechanisms in tumor cells.

Chemical entity   Protein/Gene entity

CPR:4 (Down regulator, Inhibitor)

Example 2: Result showed that **vitamin C** treatment resulted in markedly reduced **c-Rel** nuclear translocation.

Chemical entity   Protein/Gene entity

CPR:5 (Agonist)

Example 3: We previously reported that tricyclic antidepressants act as agonists at distinct opioid receptors.

Chemical entity   Protein/Gene entity

CPR:6 (Antagonist)

Example 4: **GDC-0152** is a small-molecule drug that triggers tumor cell apoptosis by selectively antagonizing **IAPs**.

Chemical entity   Protein/Gene entity

CPR:9 (Substrate/ Product of)

Example 5: These results suggest that **ZIP8** plays a pivotal role in the transport and toxicity of **Cd(2+)** and Mn(2+) in RBL-2H3 cells.

Protein/Gene entity   Chemical entity

CPR:4 (Down regulator, Inhibitor)   Chemical entity

Example 6: In complementary RNA (cRNA)-injected Xenopus laevis oocytes, **rifamycin SV** (10 micromol/L) cis-inhibited human organic anion transporting polypeptide C (SLC21A6) (OATP-C), human organic anion transporting polypeptide 8 (SLC21A8) (OATP8), human organic anion transporting polypeptide B (SLC21A9) (OATP-B), and human organic anion transporting polypeptide A (**SLC21A3**) (OATP-A) mediated BSP uptake by 69%, 79%, 89%, and 57%, respectively, as compared with uptake into control oocytes.
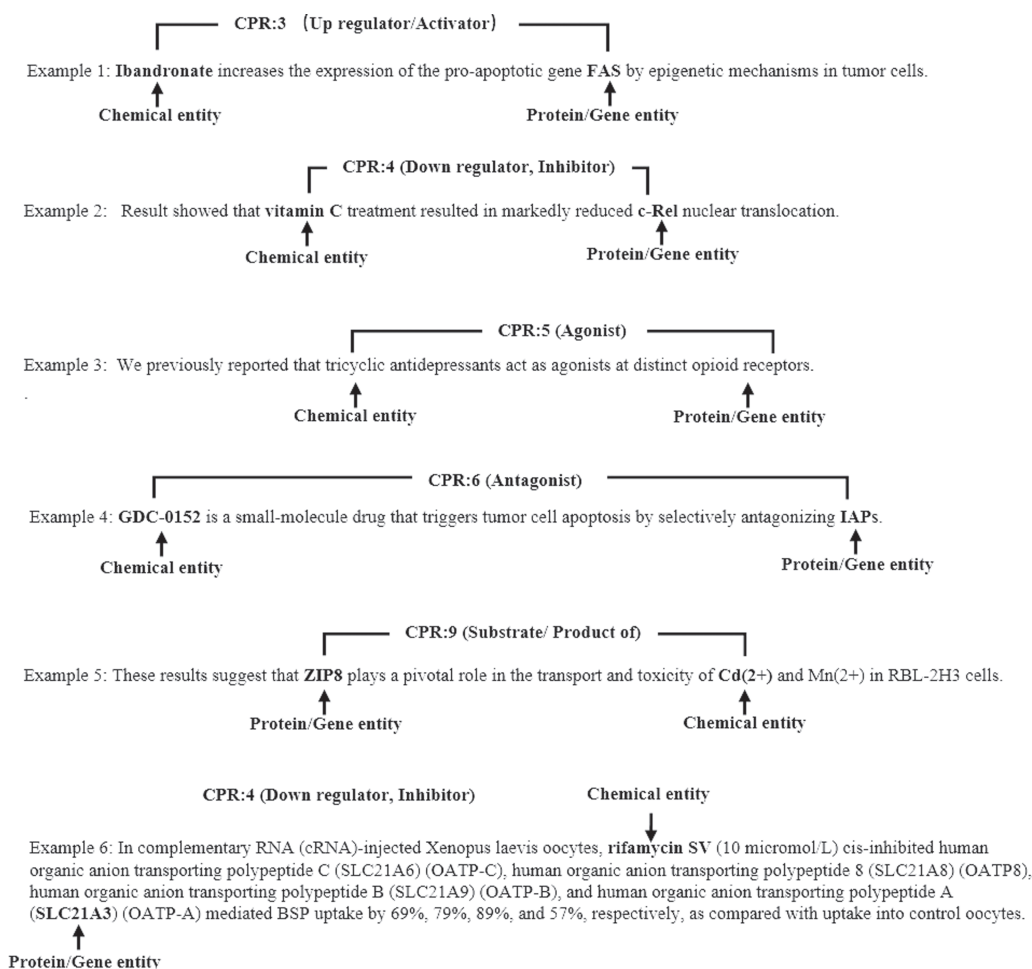
Protein/Gene entity

Figure 2. The illustrative examples of CPR classes.

Similarly, the backward language model calculates the probability of the word w_k based on the behind words $\{w_{k+1}, w_{k+2}, \ldots, w_n\}$ of w_k in S as follows:

$$P_{backward}(w_1, w_2, \ldots, w_n)$$
$$= \prod_{k=1}^{n} p(w_k | w_{k+1}, w_{k+2}, \ldots, w_n). \qquad (2)$$

A bidirectional language model combines the forward and backward language models and jointly maximizes the log likelihood as follows:

$$\sum_{k=1}^{n} (log p(w_k | w_1, w_2, \ldots, w_{k-1})$$
$$+ log p(w_k | w_{k+1}, w_{k+2}, \ldots, w_n)). \qquad (3)$$

ELMo representation is a function of a combination of the intermediate layer representations in the bidirectional language model. Therefore, ELMo can learn the different representation vector of each word in different sentences. More details about ELMo can be found in the study (18).

In our experiments, we employ the ELMo module from TensorFlow Hub.

Bi-LSTMs model

The LSTM model is currently one of the most powerful RNN models, which has been successfully applied in many NLP. Compared with traditional RNN models, the LSTM model employs the gate mechanism to solve the vanishing gradient problem (25). The LSTM model is a time sequential model and is explicitly designed to remember the information for long time periods. Therefore, the LSTM model is suitable to capture the long-term dependency feature in NLP tasks. At the time step t, each LSTM unit calculates the input word x_t, the previous hidden state h_{t-1} and the memory cell c_{t-1} to generate the current hidden state h_t and memory cell c_t (25). The current hidden state h_t and memory cell c_t can be calculated based on the equations (4)~(9). W_*, U_* and b_* denote weight and bias parameters of the LSTM units, and $\odot$ denotes element-wise
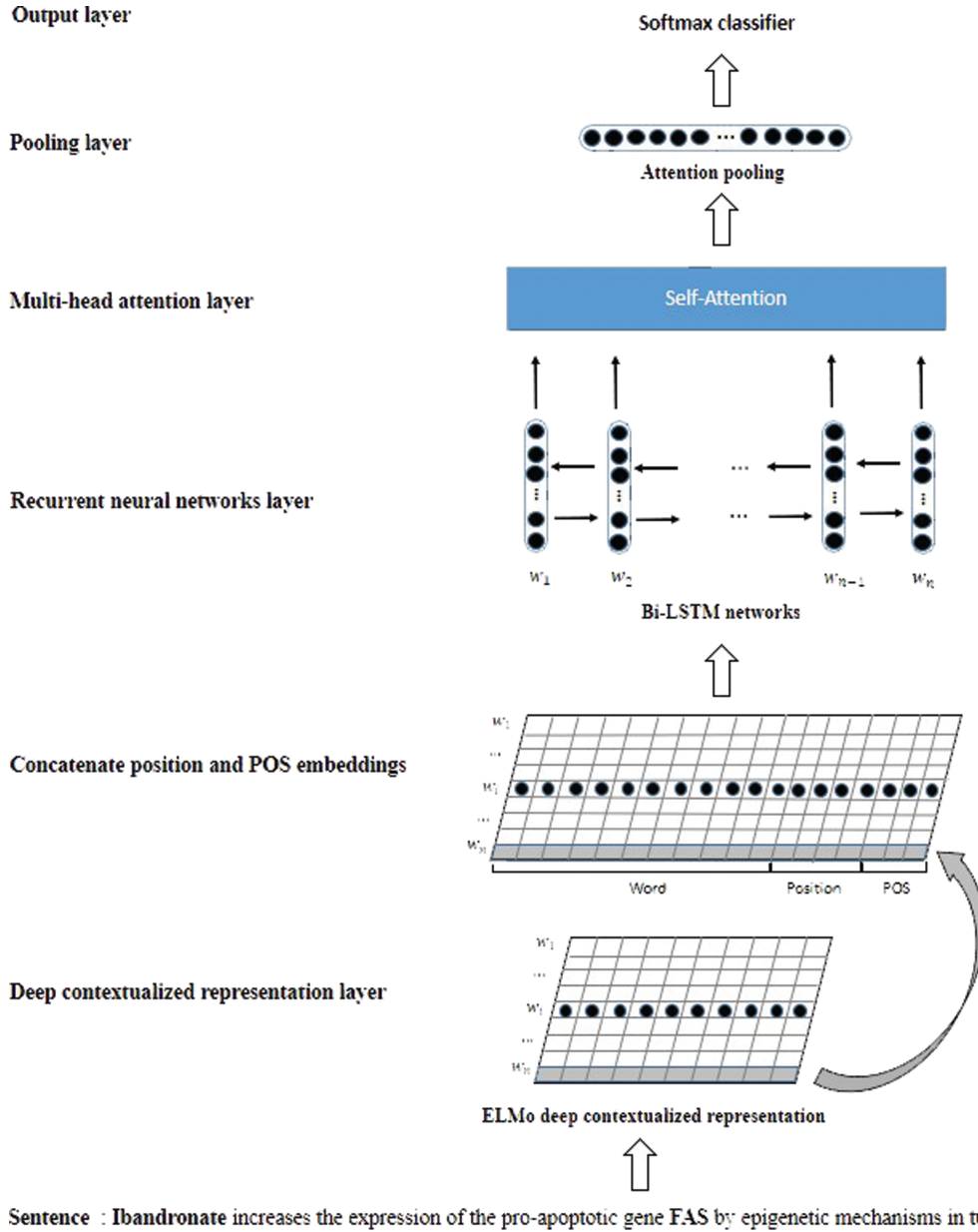
Figure 3. The schematic overview of our model.

multiplication. f_t and o_t are the state of the current forget and output gate, respectively.

$$f_t = sigmoid\left(W_f x_t + U_f h_{t-1} + b_f\right) \qquad (4)$$

$$o_t = sigmoid\left(W_o x_t + U_o h_{t-1} + b_o\right) \qquad (5)$$

$$g_t = \tanh\left(W_g x_t + U_g h_{t-1} + b_g\right) \qquad (6)$$

$$i_t = sigmoid\left(W_i x_t + U_i h_{t-1} + b_i\right) \qquad (7)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \qquad (8)$$

$$h_t = o_t \odot \tanh\left(c_{t-1}\right) \qquad (9)$$

The Bi-LSTMs model combines the forward LSTM and backward LSTM. Given h_t^f and h_t^b denote the hidden state of the forward LSTM and backward LSTM, the final hidden state of the Bi-LSTMs will be concatenated into $h_t = h_t^f \parallel h_t^b$. The Bi-LSTMs model can deal with the input sequence from the two-way approach and capture more comprehensive features.

Multihead attention

The Bi-LSTMs layer can effectively and automatically learn the latent features from the input sequences. However, only a small part of these latent features are crucial in

the CPI extraction. In our model, multihead attention (21) is employed to further emphasize the relatively important features by adjusting the weights. The intuition behind the multihead attention is that applying the attention multiple times may learn more important features than single attention. In short, the attention mechanism calculates the output based on the query and a set of key–value pairs. The multihead attention is also based on query, key and value matrixes that are denoted as $Q, K, V \in \mathbb{R}^{n \times d}$. In our study, we used the multihead self-attention to deal with the output of Bi-LSTMs. The self-attention is a special case of multihead attention, which only requires a single input.

Given $X \in \mathbb{R}^{n \times d}$ denotes the input sequence matrix, the Q, K and V will be generated by applying linear projections. Instead of the standard additive attention mechanism (26), the multihead attention uses dot-product attention to increase the parallel computation as follows:

$$Attention\,(Q, K, V) = softmax\left(QK^T/\sqrt{d}\right) V \qquad (10)$$

where $\sqrt{d}$ is the scaling factor. The key point of the multihead attention is employing the above attention multiple times. If the multihead attention contains h heads, the k-th head attention M_k can be calculated as follows:

$$M_k\,(Q, K, V) = Attention\left(QW^Q, KW^K, VW^V\right) \qquad (11)$$

where $W^Q, W^K, W^V \in \mathbb{R}^{n \times d/h}$. The final multihead attention M is the concatenation of $\{M_1, M_2, \ldots, M_h\}$.

$$M\,(Q, K, V) = Concat\,(M_1, M_2, \ldots, M_h)\, W^O \qquad (12)$$

where $W^O \in \mathbb{R}^{d \times d}$. Thus, the output of the multihead attention $M(Q, K, V)$ is a matrix of $\mathbb{R}^{n \times d}$.

Classification and training

In the pooling layer, the attention pooling (22) is applied to map the multihead attention matrix to the sentence vector representation. The 'softmax' function is used in the output layer to implement the detection and classification of CPIs.

In the experiments, our model is implemented by the Keras with the TensorFlow backend. We chose the categorical cross-entropy as the object function and use RMSProp to optimize the proposed model. A dropout mechanism was employed before the contextualized representation layer and output layer to alleviate the overfitting of the neural networks model (27). The hyperparameters used in our experiments are listed as follows. The hidden unit number of forward and backward LSTM is both 300. The mini-batch size is set as 64. The learning rate of RMSProp is set as 0.001. The dropout rate before the embedding layer

and output layer is both set as 0.5. The early stopping strategy (28) is used to choose the number of epochs on the validation set.

Results and discussion

Datasets and evaluation metrics

The ChemProt corpus (13) is the major dataset for CPI extraction, which was released on the BioCreative VI share task. In Table 1, the ChemProt corpus is annotated for 10 relation classes, but only 5 relation classes (CPR:3, CPR:4, CPR:5, CPR:6 and CPR:9) were evaluated. To maintain the same evaluation with other studies, we also focused on the five-type relation classes. The statistics of the ChemProt corpus are listed in Table 2.

In the experiments, we combined the training and development sets as a whole training set. The validation set was randomly chosen from the training samples with a 10% rate. We used the validation to choose the parameters of our model. The test set was just used for evaluating the performance of our model.

The *F*-score, 'precision' and 'recall' are widely used as the evaluation metrics in CPI extraction. In particular, the *F*-score is the harmonic mean of both 'precision' and 'recall', which can quantify the overall performance. Therefore, the *F*-score was chosen as the major metric in our experiments. Since the CPI extraction is a multiclass classification task, we compute the micro average to assess the overall performance (14). To reduce the potential bias, we repeated each experiment five times and reported the average *F*-score, 'precision' and 'recall'.

In the comparison experiments, we used the pretrained biomedical word embedding (29), which is available at https://github.com/cambridgeltl/BioNLP-2016. For ELMo model, we used pretrained ELMo embeddings from TensorFlow hub (https://tfhub.dev/google/elmo).

Experimental results on ChemProt corpus

We first evaluated the effectiveness of different input representations of our method. In this experiment, we used the same Bi-LSTMs and changed the input representations. The comparison performance of different input representations are listed in Table 3. To focus on input representation evaluation, all models in Table 3 do not use attention strategy.

- 'Word': the input representation of the model is word embedding.
- 'Word + Position': the input representation of the model is the concatenated of word embedding and position embedding.

Table 3. The effect of the input representation on performance

Input representation	Train vs validation			Train vs test		
	Precision	Recall	*F*-score	Precision	Recall	*F*-score
Word	0.554	0.585	0.569	0.492	0.624	0.55
Word + Position	0.524	0.633	0.573	0.511	0.643	0.569
Word + Position + POS	0.601	0.596	0.598	0.586	0.57	0.578
Context	0.592	0.641	0.616	0.596	0.628	0.611
Context + Position	0.627	0.643	0.635	0.578	0.677	0.624
Context + Position + POS	0.629	0.66	0.644	0.621	0.64	0.629

Notes: 'Word', 'Position', 'POS' and 'Context' denote word embedding, position embedding, POS embedding and deep context representation, respectively. 'Train vs validation' and 'Train vs test' denote the results on validation and test sets, respectively. Attention mechanism is not used in this experiment.

- 'Word + Position + POS': the input representation of the model is the concatenated of word embedding, position embedding and POS embedding.
- 'Context': the input representation of the model is deep context embedding.
- 'Context + Position': the input representation of the model is the concatenated of deep context embedding and position embedding.
- 'Context + Position + POS': the input representation of the model is the concatenated of deep context embedding, position embedding and POS embedding.

In Table 3, it can be seen that the Bi-LSTMs model achieved an *F*-score of 0.55 when only using pretrained word embedding as the input representation. Both position embedding and POS embedding are helpful to improve the performance in CPI extraction. When combining the word, position and POS embedding, the *F*-score was improved from 0.55 to 0.578. Compared to the word embedding, deep context representation ELMo greatly improved the *F*-score from 0.578 to 0.629. The results suggest that the ELMo can generate a more comprehensive representation of the words from the intermediate layer based on the sentence contexts. This makes the ELMo outperform pretrained word embedding for the performance of CPI extraction. Moreover, combining the position and POS embed-

dings with deep context representation can further improve the performance. Overfitting is a common issue in this domain. We also report the performance measures on the validation set. Compared with the results on the validation set, the *F*-score only slightly decrease (≤0.015). This indicates that the models do not suffer much from overfitting.

Then, we evaluated the effectiveness of the multihead attention in CPI extraction. In this experiment, all models employed the multihead attention mechanism and used the concatenated of word embedding, position embedding and POS embedding as the input representation. Since multihead self-attention was employed, the dimensions of the query *Q*, key *K* and value *V* are the same. As shown in Table 4, we varied the number of attention heads and the dimensions of *Q*, *K* and *V*. In Table 4, we can see multihead attention can effectively improve the performance of CPI extraction. The experimental results indicate that multihead attention can combine the important features from the different heads to construct a comprehensive feature representation. We also noticed that the *F*-score ranged from 0.646 to 0.659 when setting different head numbers. When the number of heads is set too small or too large, the performance will drop off. Overall, our model achieved the highest *F*-score of 0.659 when the number of heads was

Table 4. The effect of the attention heads and the dimensions on performance

Heads	Dimensions	Train vs validation			Train vs test		
		Precision	Recall	*F*-score	Precision	Recall	*F*-score
2	300	0.692	0.619	0.653	0.663	0.634	0.648
4	150	0.685	0.634	0.659	0.691	0.62	0.653
6	100	0.697	0.642	0.668	0.706	0.618	0.659
12	50	0.69	0.651	0.67	0.696	0.622	0.657
20	30	0.674	0.648	0.661	0.67	0.629	0.649

Notes: 'Heads' and 'Dimensions' denote the number of heads and the dimensions of *Q*, *K* and *V*, respectively. 'Train vs validation' and 'Train vs test' denote the results on validation and test sets, respectively. The input representation is the combination of deep context embedding, position embedding and POS embedding.
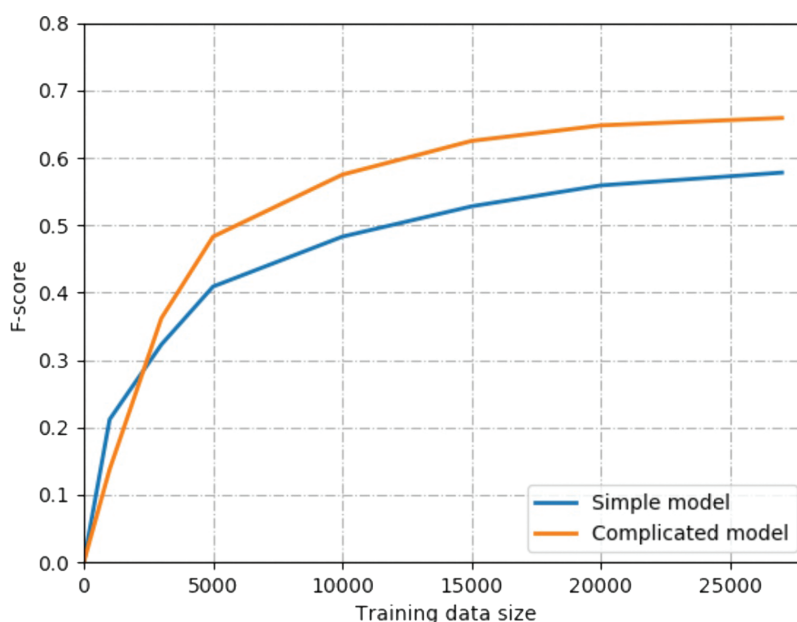
Figure 4. The learning curve of two models on ChemProt corpus. The simple model contains the combination input of the word, position and POS embeddings and Bi-LSTMs. The complicated model contains the combination input of deep context, position and POS embeddings, Bi-LSTMs and multihead attention mechanism.

set as 6, which outperformed the *F*-score of 0.629 without using multihead attention.

Next, we evaluated the performance with different sizes of training data. Generally, more labeled training samples are always helpful for the supervised learning model. However, in the biomedical domain, annotating labeled data is expensive and time-consuming. The key point is how many training data are required for competitive performance on biomedical relation extraction. Since we combined the training and development sets of the ChemProt corpus as a whole training set, the labeled training samples are ~27 000. We experimented with the differently labeled training samples that ranged from 1000 to 27 071. The evaluation results of the two models are shown in Figure 4. The simple model contains the combination input of the word, position and POS embeddings and Bi-LSTMs. The complicated model contains the combination input of deep context, position and POS embeddings, Bi-LSTMs and multihead attention mechanism. For the complicated model, the training sample size ranges from 1000 to 27 071, with a corresponding *F*-score of 0.128 to 0.659. Similarly, the corresponding performance of the simple model was increased from 0.222 to 0.578 in *F*-score. We made the observation that the performance increase slowed down when the number of training samples increased. For example, the corresponding performance of the complicated model increased from 0.128 to 0.483 when the training sample size increased from 1000 to 5000. However, the performance increase of the complicated model is only 0.011 when the training sample size increases from 20 000 to 27 071. Similar trends

can be found for the simple model. Our results show that both models with ~5000 training samples can achieve >70% of the best performance. In addition, we can see the simple model outperforms the complicated model in *F*-score when using 1000 or 3000 training samples. The main reason is that the complicated model has much more parameters need to train than the simple model because of the ELMo representation and multihead attention layers. Therefore, the complicated model requires more training samples to train than the simple model. When the number of training samples is >5000, the complicated model can outperform the simple model significantly.

Analysis of the word-level attention weights

Figure 5 shows the attention weight distribution of some instances from ChemProt test set. Darker background color (dark red) on the word indicates higher attention weights. It can be seen that the attention mechanism can highlight some important keywords, such as 'agonist', 'inhibiting' and 'reduced'. These keywords are greatly helpful to classify the semantic relation type.

Table 5 gives the top 5 attention keywords in each relation type among the ChemProt test set. The recent study (22) has suggested that CPR:4, CPR:5 and CPR:6 have high concentration of keywords. We also found similar results. In Table 5, we can see, in CPR:4 (downregulator and inhibitor), all the top 4 key words are the variations of 'inhibitor'. In CPR:5 (agonist) and CPR:6 (antagonist), the top words concentrate in the variation of 'agonist'
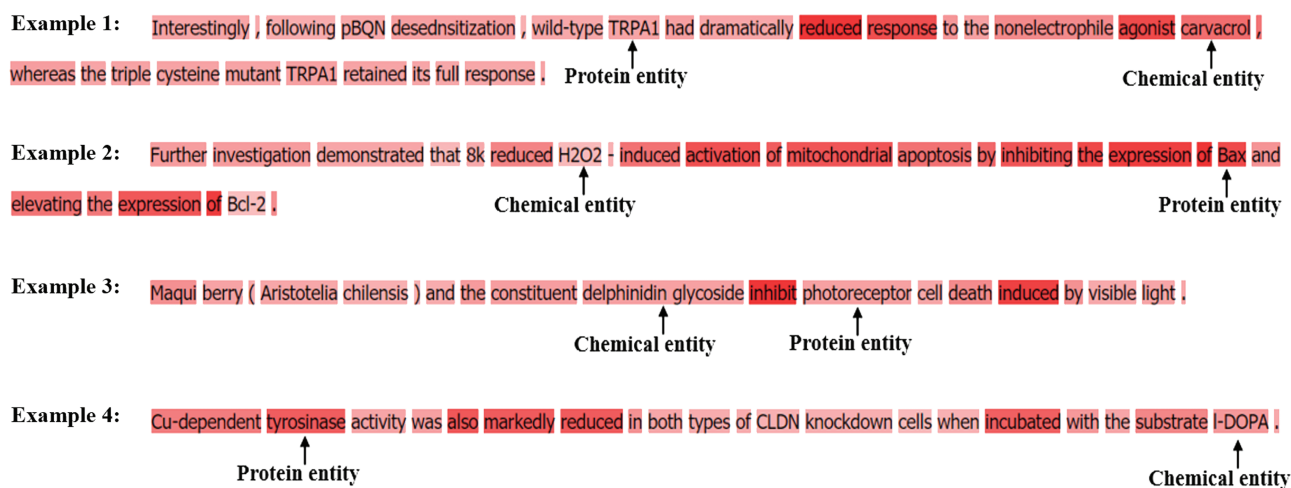
Figure 5. Examples of attention weight distribution.

Table 5. Top 5 attention keywords in each relation type

CPR:3 (upregulator and activator)	CPR:4 (downregulator and inhibitor)	CPR:5 (agonist)	CPR:6 (antagonist)	CPR:9 (substrate and product of)
Expression	Inhibitor	Agonist	Antagonist	Metabolism
Increased	Inhibition	Selective	Antagonists	Catalyzes
Induced	Inhibitors	Activity	Selective	Metabolized
Activation	Inhibited	Agonists	Receptor	Uptake
Activity	Cells	Antagonist	Binding	Enzyme

and 'antagonist', respectively. These top words such as 'inhibitor', 'agonist' and 'antagonist' are the strong indicator of CPR:4, CPR:5 and CPR:6. For CPR:3 (upregulator and activator) and CPR:9 (substrate and product of), our model identified deeper semantic variants such as 'increased', 'expression' and 'enzyme'.

Performance comparison on ChemProt corpus

We compared our method with other state-of-the-art methods on ChemProt corpus in Table 6. Warikoo *et al.* (30) employed a linguistic interaction pattern learning method to capture the CPI and achieved an *F*-score of 0.526 on ChemProt corpus. Lung *et al.* (31) used a three-stage model to integrate the semantic and dependency graph features and achieved an *F*-score of 0.567. Liu *et al.* (22) applied the GRU model with attention pooling and achieved an *F*-score of 0.527. Similarly, Corbett *et al.* (15) also used a Bi-LSTMs model with pretrained LSTM layers to achieve a high *F*-score of 0.615. Peng *et al.* (14) proposed an ensemble method to combine three system results including SVM, CNN and Bi-LSTMs. This method achieved an *F*-score of 0.641, which was the top rank in the BioCreative VI ChemProt share task.

In Table 6, neural network-based methods achieved highly competitive performance in the CPI extraction task. Compared with other methods, our model effectively integrated the deep context representation, Bi-LSTMs and multihead attention and achieved the highest *F*-score of 0.659. We noticed that some studies (14, 15, 22) also applied Bi-LSTMs in CPI extraction. The Bi-LSTMs model can automatically learn the long-term latent features from the sentence. For CPI extraction, some sentences are long and complicated. Moreover, some chemical and protein entities are in different clauses. Thus, it is difficult to learn enough features for Bi-LSTMs to distinguish and classify the candidate CPI. To solve this problem, we applied the deep context representation instead of word embedding to

Table 6. Performance comparison with other methods on ChemProt corpus

Methods	Precision	Recall	*F*-score
Warikoo *et al.* (30)	0.592	0.474	0.526
Lung *et al.* (31)	0.632	0.512	0.567
Liu *et al.* (22)	0.574	0.487	0.527
Corbett *et al.* (15)	0.561	0.678	0.615
Peng *et al.* (14)	0.727	0.574	0.641
Our method	0.706	0.618	0.659

Notes: the input representation is the combination of deep context embedding, position embedding and POS embedding. The number of the attention heads and dimensions are set as 6 and 100, respectively.

generate the comprehensive sentence representation, which provided much more comprehensive information to the Bi-LSTMs. Furthermore, we employed a multihead attention layer to effectively enhance the important distinguishing features based on the Bi-LSTMs output. Therefore, both the deep context representation and multihead attention strategies were helpful to improve the performance in CPI extraction. Our experiments were performed on NVIDIA GPU GeForce GTX Titan Xp with 12 GB GDDR5X memory. The training time for one epoch of our model was ∼317 s.

Overall, our method took advantage of the deep context representation and multihead attention strategies to achieve state-of-the-art performance on the ChemProt corpus.

Performance breakdown and error analysis

Table 7 gives the performance breakdown of each CPI type. It can be seen that the performance of different CPI type vary significantly. Our model achieved a high *F*-score of 0.725 and 0.718 on CPR:6 (antagonist) and CPR:4 (down-regulator and inhibitor), respectively. On the contrary, the *F*-score on CPR:9 (substrate and product of) is only 0.501. This indicates that it is the most difficult for our model to accurately classify CPR:9 type CPIs. Table 8 shows the confusion matrix for our model on the test set. The *x*-axis is the predicted label by our model, and the *y*-axis is the gold standard label. In Table 8, we can see that the major challenge is nonrelations being mistaken for relations and vice versa. Besides that, there is something of a problem with CPR:3 (upregulator and activator) being mistaken for CPR:4 (downregulator and inhibitor). This suggests that accurately distinguishing between CPR:3 and CPR:4 is another challenge for our model.

In addition, we also manually analyzed what sentences lead to false negatives. In Figure 6, we gave some examples of false negatives. The chemical and protein entities are in bold. We noted that the passive structure is a frequent cause of false negatives. For example, in FN1, the chemical entity 'estradiol' is in passive. In this case, our model misclassified

Table 7. Performance breakdown of our model on the test set

Label	Support	Precision	Recall	*F*-score
CPR:3	664	0.662	0.539	0.594
CPR:4	1661	0.704	0.732	0.718
CPR:5	194	0.737	0.593	0.657
CPR:6	281	0.759	0.694	0.725
CPR:9	643	0.735	0.379	0.501

Notes: the input representation is the combination of deep context embedding, position embedding and POS embedding. The number of attention heads and dimensions are set as 6 and 100, respectively.

Table 8. Confusion matrix for our model on the test set

Gold	False	CPR:3	CPR:4	CPR:5	CPR:6	CPR:9
False	10 267	164	415	32	51	84
CPR:3	220	358	80	3	2	1
CPR:4	427	14	1216	0	1	3
CPR:5	66	2	3	115	8	0
CPR:6	77	0	3	6	195	0
CPR:9	386	3	10	0	0	244

Notes: the input representation is the combination of deep context embedding, position embedding and POS embedding. The number of attention heads and dimensions are set as 6 and 100, respectively.

the relation between 'ERÎ±' and 'estradiol' as false. Another frequent cause of false negatives is the sentence is long and complicated. For example, in FN3, the chemical and protein entities are in different clauses and the sentence is relatively complicated. In this case, our model failed in identifying the relation between 'Fas-associated death domain-containing protein' and 'paclitaxel'. In the future plan, more effort should be made on how to identify CPIs from passive structures and long sentences accurately.

Experimental results on DDI 2013 corpus

We also evaluated our model on DDI 2013 corpus (6, 32) that contains four DDI types: 'Advice', 'Effect', 'Mechanism' and 'Int'. Table 9 shows the detailed statistics of DDI 2013 corpus. In this experiment, we used the same hyperparameters. Since the recent studies (8, 24) have suggested that the position and POS embedding can improve the performance of DDI extraction, we concatenated the word or deep context embedding with position and POS embedding as the input representation in this experiment. We mainly focused on the effectiveness of context representation and multihead attention strategy in the ablation study. Table 10 gives the ablation study results on the DDI 2013.

- 'Word + Position + POS': the input representation of the model is the concatenated of word embedding, position embedding and POS embedding. Attention mechanism is not used in this model.
- 'Context + Position + POS': the input representation of the model is the concatenated of deep context embedding, position embedding and POS embedding. Attention mechanism is not used in this model.
- 'Context + Position + POS + Attention': the input representation of the model is the concatenated of deep context embedding and position embedding. The multihead attention strategy is employed in this model. The number of attention heads and dimensions are set as 6 and 100, respectively.

In Table 10, it can be seen that both context representation and multihead attention strategy are also helpful on

Passive structures

CPR:3 (Upregulator, Activator)

FN1: These results indicate that membrane **ERÎ±** levels in N-38 neurons are dynamically autoregulated by **estradiol**.

Protein entity   **Chemical entity**

CPR:4 (Downregulator,Inhibitor)

FN2: The selective **CaMKIIÎ±** inhibitor KN-62 reversed the blockade produced by **ionomycin** and K(+)-depolarization.

Protein entity   **Chemical entity**

Long sentence and multiple subordinate clauses

CPR:3 (Upregulator, Activator)

Protein entity

FN3: We have previously demonstrated that phosphorylation of **Fas-associated death domain-containing protein** (FADD) at 194 serine through c-jun NH2-terminal kinase (JNK) activation sensitizes breast cancer cells to chemotherapy through accelerating cell cycle arrest at G2/M, and that Bcl-2 phosphorylation downstream of JNK/FADD plays an important role in cell growth suppression by **paclitaxel**.

Chemical entity

CPR:4 (Downregulator,Inhibitor)

Chemical entity

FN4: In HaCaT cells, buthanol and **ethylacetate** fractions of 80% methanol C. fragile extract (CFB or CFE) and a single compound, clerosterol (CLS) isolated from CFE attenuated UVB (60mJ/cm(2))-induced cytotoxicity and reduced expression of pro-inflammatory proteins including cyclooxygenase-2 (COX-2), inducible nitric oxide synthase (iNOS), and tumor necrosis factor-Î± (**TNF- Î±**).

Protein entity

Figure 6. Examples of false negatives.

Table 9. The statistics of the DDI extraction 2013 corpus

Corpus	Advice	Effect	Mechanism	Int	Negative
Training set	826	1687	1319	188	23 772
Test set	221	360	302	96	4737
Total	1047	2047	1621	284	28 554

Table 10. The ablation study on DDI 2013 corpus

Models	Precision	Recall	*F*-score
Word + Position + POS	0.774	0.62	0.688
Context + Position + POS	0.787	0.658	0.716
Context + Position + POS + Attention	0.782	0.674	0.724

Notes: 'Word', 'Position', 'POS', 'Context' and 'Attention' denote word embedding, position embedding, POS embedding, deep context representation and attention mechanism, respectively.

Table 11. Performance comparison with other methods on DDI 2013 corpus

Methods	Precision	Recall	*F*-score
Zhao *et al.* (8)	0.725	0.651	0.686
Raihani and Laachfoubi (33)	0.737	0.687	0.711
Quan *et al.* (34)	0.76	0.653	0.702
Sahu and Anand (35)	0.734	0.697	0.715
Zhang *et al.* (12)	0.75	0.725	0.737
Our method	0.782	0.674	0.724

Notes: the input representation is the combination of deep context embedding, position embedding and POS embedding. The number of attention heads and dimensions are set as 6 and 100, respectively.

DDI 2013 corpus. In particular, compared to word embedding, the context representation embedding improved the *F*-score from 0.688 to 0.716.

In Table 11, we compared our model with other state-of-the-art methods on DDI 2013 corpus. Zhao *et al.* (8) employed CNN model to extract DDIs and achieved an *F*-score of 0.686. Raihani and Laachfoubi (33) constructed rich features and employed feature-based method to extract DDIs and achieved a high *F*-score of 0.711. Quan *et al.* (34) used multichannel CNN model to extract DDIs, which can effectively integrate multiple input representation for

DDI extraction task. Sahu and Anand (35) applied Bi-LSTM model and attention pooling to extract DDIs, which achieved a high *F*-score of 0.711. Zhang *et al.* (12) proposed a hybrid model to combine CNNs and Bi-LSTM for DDI extraction task. To boost the performance, the hybrid model input not only the token sequence of sentences but also the shortest dependency path between the two drug entities. The hybrid model achieved the highest *F*-score of 0.737 on DDI 2013 corpus. All these methods focused on the DDI extraction task. Compared with other methods, our model can achieve competitive performance on DDI 2013 without fine-tuning hypermeters. This indicated that our model can be applied to other biomedical relation extraction task.

Conclusion

Accurately detecting and extracting CPIs from the literature is a crucial task in the biomedical domain. However, the best performance of CPI extraction is ~0.64 in *F*-score. Both the deep context representation and multihead attention strategies are the most recent advantages of deep learning, which could improve the performance of CPI extraction. The deep context representation can effectively generate the sentence representation according to the sentence contexts. The multihead attention mechanism learns the important features from different heads and has the ability of generating more comprehensive and distinguished feature representation. In this work, we proposed a neural networks-based method to integrate the deep context representation, Bi-LSTMs and multihead attention in CPI extraction. The proposed method was evaluated on the recent ChemProt corpus. The results show that both deep context representation and multihead attention improve the performance in CPI extraction. It is encouraging to see that the proposed model achieved the highest performance of 0.659 in *F*-score on the ChemProt corpus. The experimental results on DDI 2013 also suggest our methods can be applied to other biomedical relation extraction tasks.

Generally, supervised learning methods depend on sufficient labeled training data. However, annotating training data is expensive and time-consuming, especially so in biomedicine as domain knowledge is required. In the future, we will explore the effectiveness of semi-supervised learning or transfer learning in CPI extraction.

References

1. Kringelum,J., Kjaerulff,S.K., Brunak,S. *et al.* (2016) ChemProt-3.0: a global chemical biology diseases mapping. *Database (Oxford)*, **2016**, 1–7, bav123.
2. Airola,A., Pyysalo,S., Björne,J. *et al.* (2008) All-paths graph kernel for protein–protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, **9**, S2.
3. Miwa,M., Sætre,R., Miyao,Y. *et al.* (2009) A rich feature vector for protein–protein interaction extraction from multiple corpora. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Vol. 1. Association for Computational Linguistics, Singapore, 121–130.
4. Kim,S., Yoon,J., Yang,J. *et al.* (2010) Walk-weighted subsequence kernels for protein–protein interaction extraction. *BMC Bioinformatics*, **11**, 107.
5. Zhang,Y., Lin,H., Yang,Z. *et al.* (2012) A single kernel-based approach to extract drug–drug interactions from biomedical literature. *PLoS One*, **7**, e48901.
6. Segura-Bedmar,I., Martínez,P. and Herrero-Zazo,M. (2014) Lessons learnt from the DDIExtraction-2013 shared task. *J. Biomed. Inform.*, **51**, 152–164.
7. Segura-Bedmar,I., Martínez,P. and de Pablo-Sánchez,C. (2011) A linguistic rule-based approach to extract drug–drug interactions from pharmacological documents. *BMC Bioinformatics*, **12**, S1.
8. Zhao,Z., Yang,Z., Luo,L. *et al.* (2016) Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. *Bioinformatics*, **32**, 3444–3453.
9. Li,F., Zhang,M., Fu,G. *et al.* (2017) A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*, **18**, 198.
10. Wang,J., Zhang,J., An,Y. *et al.* (2016) Biomedical event trigger detection by dependency-based word embedding. *BMC Med. Genomics*, **9**, 45.
11. Rios,A., Kavuluru,R. and Lu,Z. (2018) Generalizing biomedical relation classification with neural adversarial domain adaptation. *Bioinformatics*, **34**, 2973–2981.
12. Zhang,Y., Lin,H., Yang,Z. *et al.* (2018) A hybrid model based on neural networks for biomedical relation extraction. *J. Biomed. Inform.*, **81**, 83.
13. Krallinger,M., Rabal,O. and Akhondi,S.A. (2017) Overview of the BioCreative VI chemical–protein interaction track. In: *Proceedings of the BioCreative VI Workshop*. Bethesda, MD, USA, pp. 141–146.
14. Peng,Y., Rios,A., Kavuluru,R. *et al.* (2018) Extracting chemical–protein relations with ensembles of SVM and deep learning models. *Database (Oxford)*, **2018**, 1–9, bay073.
15. Corbett,P. and Boyle,J. (2018) Improving the learning of chemical–protein interactions from literature using transfer learning and specialized word embeddings. *Database (Oxford)*, **2018**, 1–10, bay066.
16. Mikolov,T., Chen,K., Corrado,G. *et al.* (2013) Efficient estimation of word representations in vector space. In: *Proceedings of International Conference on Learning Representations*. Scottsdale, Arizona, pp. 1–12.
17. Pennington,J., Socher,R. and Manning,C. (2014) Glove: global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532–1543.
18. Peters,M.E., Neumann,M., Iyyer,M. *et al.* (2018) Deep contextualized word representations. In: *Proceedings of NAACL-HLT 2018*, New Orleans, USA, pp. 2227–2237.
19. Du,X., Shao,J. and Cardie,C. (2017) Learning to ask: neural question generation for reading comprehension. In: *Proceedings of Association for Computational Linguistics*, 2017, Vancouver, Canada, pp. 1342–1352.
20. Chen,Q., Zhu,X., Ling,Z.H. *et al.* (2017) Recurrent neural network-based sentence encoder with gated attention for natural language inference. In: *Proceedings of the 2nd Workshop on Evaluating Vector-Space Representations for NLP*, Copenhagen, Denmark, 7–11 September 2017, pp. 36–40.

21. Vaswani,A., Shazeer,N., Parmar,N. *et al.* (2017) Attention is all you need. *Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, USA, pp. 5998–6008.

22. Liu,S., Shen,F., Komandur Elayavilli,R. *et al.* (2018) Extracting chemical–protein relations using attention-based neural networks. *Database (Oxford)*, **2018**, 1–12, bay102.

23. Verga,P., Strubell,E. and McCallum,A. (2018) Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In: *Proceedings of NAACL-HLT*, 2018, New Orleans, USA, pp. 872–884.

24. Zhang,Y., Zheng,W., Lin,H. *et al.* (2018) Drug–drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths. *Bioinformatics*, **34**, 828–835.

25. Hochreiter,S. and Schmidhuber,J. (1997) Long short-term memory. *Neural Comput.*, **9**, 1735–1780.

26. Bahdanau,D., Cho,K. and Bengio,Y. (2014) Neural machine translation by jointly learning to align and translate. *Comput. Sci.*, **1409**, 0473.

27. Srivastava,N., Hinton,G., Krizhevsky,A. *et al.* (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**, 1929–1958.

28. Prechelt,L. (1998) Automatic early stopping using cross validation: quantifying the criteria. *Neural Netw.*, **11**, 761–767.

29. Chiu,B., Crichton,G., Korhonen,A. *et al.* (2016) How to train good word embeddings for biomedical NLP. In: *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, Berlin, Germany, pp. 166–174.

30. Warikoo,N., Chang,Y.-C. and Hsu,W.-L. (2018) LPTK: a linguistic pattern-aware dependency tree kernel approach for the BioCreative VI CHEMPROT task. *Database (Oxford)*, **2018**, 1–21, bay108.

31. Lung,P.Y., Zhao,T. and He,Z. (2017) Extracting chemical protein interactions from literature. In: *Proceedings of 2017 BioCreative VI Workshop, October 2017, Maryland, USA. BioCreative VI*, pp. 160–163.

32. Herrero-Zazo,M., Segura-Bedmar,I., Martínez,P. *et al.* (2013) The DDI corpus: an annotated corpus with pharmacological substances and drug–drug interactions. *J. Biomed. Inform.*, **46**, 914–920.

33. Raihani,A. and Laachfoubi,N. (2016) Extracting drug–drug interactions from biomedical text using a feature-based kernel approach. *J. Theor. Appl. Inform. Technol.*, **92**, 109.

34. Quan,C., Hua,L., Sun,X. *et al.* (2016) Multichannel convolutional neural network for biological relation extraction. *Biomed Res. Int.*, **2016**, e1850404.

35. Sahu,S.K. and Anand,A. (2018) Drug–drug interaction extraction from biomedical texts using long short-term memory network. *J. Biomed. Inform.*, **86**, 15–24.