Extracting Chemical-protein interactions via bi-directional long short-term memory network

Wei Wang[1], Xi Yang[1], Yuting Xing[1], Chengkun Wu[1], Zhuo Song[2]
[1]School of Computer Science, National University of Defense Technology, Changsha, China
[2]Genetalks Biotech. Co.,Ltd., Beijing, China

***Abstract*—Understanding chemical-protein interactions (CPI) has been of great importance to drug discovery, precision medicine and basic biomedical research. It is a time-consuming and laborious task to annotate CPIs from numerous unstructured texts. We can employ automated methods to improve the efficiency of this task. In this work, we propose a CPI extraction method based on the bi-directional long short-term memory network (a specific type of deep neural network), which does not require a complicated feature engineering procedure. Our key strategy is to break each sentence into fragments according to the position of the targeted entity pair and recombine them into chunks, which can help capture the structural knowledge hidden in the sentence. More specifically, our model consists of four network layers, including a feature layer, a Bi-LSTM layer, a pooling layer and a Softmax layer. Our results demonstrate that such a structure is beneficial for effective relation information.**

Keywords—Chemical-protein interaction; bi-directional long short-term memory network; structural knowledge;

I. INTRODUCTION

In clinical medicine, chemical drugs can act as therapeutic agents by targeting at some specific receptors and altering their structures, these receptors are usually some specific proteins, such as enzymes, oncogene products, anti-bodies, and etc. Therefore, understanding chemical and protein interactions (CPI) is of great importance to drug discovery, precision medicine and basic biomedical research. Biomedical researchers have studied a great amount of associations between chemicals and proteins, published their studies in the biomedical literature and added curated knowledge to some chemical-protein interaction databases, such as the protein data bank (https://www.rcsb.org/pdb/home/home.do) and the PDSP Ki Database (https://pdsp.unc.edu/databases/kidb.php). However, these databases are far from complete as it is time and labor-consuming to keep them up-to-date manually with the sharply growing volume of biomedical literature. Automated methods can greatly improve the efficiency of CPI extraction from unstructured texts.

Previously, various approaches were developed to address similar problems like DDIs (Drug-drug interactions) and PPIs (Protein-protein interactions) extraction. These approaches can be roughly divided into two categories: rule-based methods and machine learning methods. In general, the former approaches define a set of rules to capture various forms of expressing the relationship between two entities in texts. In (1), a set of syntactic rules and domain-specific lexical rules were applied in the identification of DDIs. Corney et al. (2) applied manually engineered templates that combine lexical and semantic information to identify PPIs. Besides manually crafted rules, it is also possible to generate rules automatically . For instance, Blasco et al. (3) proposed an automated method to summarize rules from a large amount of biomedical texts, which utilized Maximal Frequent Sequences (MFS) to discover patterns and such patterns have been proved to perform well on identifying sentences that contain DDIs. Generally speaking, rule engineering approaches are hard to scale up to large document collections due to various limitations.

Machine learning methods adopt statistical models to capture relational information via training on a set of training data. Several such methods have been proposed to extract DDIs from biomedical texts. (4) established the Turku event extraction system (TEES), and it supported detecting and identifying DDIs simultaneously by constructing a multi-class support vector machine (SVM). Liu et al. (5) introduced the convolutional neural network (CNN) into the DDI extraction task.

As for PPI extraction, feature-based methods and kernel-based methods are widely used. Feature-based methods focus on designing effective features including lexical, syntactic and semantic information. (6) used Maximum Entropy models to combine diverse lexical, syntactic and semantic features for PPI extraction. Kernel-based methods are even more effective for capturing syntactic structure information, which compute the structure similarity by kernel functions. Bunescu and Mooney (7) adopted a generalized substring kernel over a mixture of words and word classes to extract PPIs from biomedical corpora as well as semantic relations from the newswire corpora. Chowdhury et al. (8) investigated the effect of mildly extended dependency trees using an un-lexicalized partial tree kernel. Recently, deep learning techniques have achieved notable results in some PPI extraction tasks (9,10). However, to the best of our knowledge, deep learning is barely seen in CPI extraction yet.

In this work, we propose a bi-directional long short-term memory network (Bi-LSTM) based model to accomplish the task of CPI extraction without complicated feature engineering. A key strategy of our work is that we split the sentence into fragments and recombine them into chunks, expecting to capture the structure knowledge hidden in the sentence. Our model consists of four network layers, including a feature layer, a Bi-LSTM layer, a pooling layer and a

Softmax layer. In the feature layer, the sentence in each instance is split into three fragments according to the position of the target chemical entity and protein. We re-organize these three fragments into two chunks and represent them with word features and position features. Here the exact words are initialized with syntax word embedding and the position features are mapped into ten bit binary vectors. Subsequently, in the Bi-LSTM layer, two separate Bi-LSTM are equipped for each chunk with the scope of better learning relation information. After that, in the pooling layer, we employ piece max pooling rather than max pooling on the encoding sequence data obtained from the Bi-LSTM layer. Lastly, all results are concatenated together, and fed to the Softmax layer for CPI classification.

II. METHODS

In this work, we propose a bi-directional long short-term memory network based model for CPI extraction. Figure 1 shows the architecture of our model and the components are described in detail in the following parts.

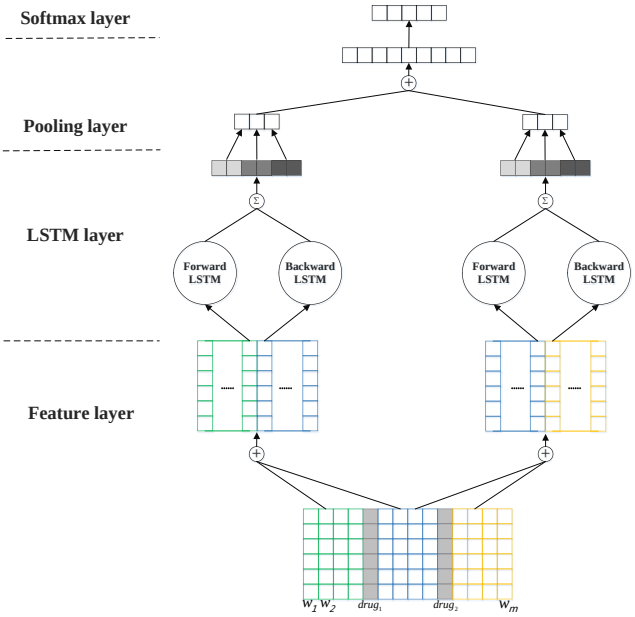

Figure 1 The framework of our model

A. Feature layer

Depending on the position of the two targeted entities, we split each instance into three fragments, named as $Fragment_1(F_1), Fragment_2(F_2), Fragment_3(F_3)$. F_1 denotes the fragment of a sentence in the front of the forward entity word, F_2 is the fragment between two targeted entities, and F_3 is the remaining fragment of the sentence behind the latter entity word. Subsequently, these three fragments are recombined into three chunks to represent the sentence. Here we combine F_1 and F_2 as $Chunk_1(Ch_1)$, denoted as $Ch_1 = F_1 + F_2$. Similar to Ch_1, F_2 and F_3 are combined into $Chunk_2(Ch_2)$. In this way, we expect to capture the structure information of the sentence.

We follow earlier researches (11,12) to characterize each word in a sentence with word features and position features. Specifically, each word in a sentence is represented with three features: word (w), $\text{Position}_1(P_1)$, and $\text{Position}_2(P_2)$, where w is the exact word, P1 and P2 are the relative distances from the current word to two targeted entities (a negative distance means backwards).

B. Bi-LSTM layer

According to previous studies, Bi-LSTM has been proved to be an excellent model in processing long sequential data, especially for text data. Thus, we employ three Bi-LSTM model to encode each chunk constructed from the previous layer, aiming to better capture effective encoding information for relation extraction.

The Bi-LSTM model is equipped with two parallel LSTM layers, forward LSTM layer and backward LSTM layer. As theoretical analysis and experimental results show that the long sequence data exits long-term dependencies problem, the LSTM model emerged. Based on the recurrent neural network architecture, a new structure of the memory block is introduced into the LSTM model to alleviate the vanishing gradient problem. More precisely, the memory block consists of a memory cell (C_t) and three multiplicative gates, including the input gate (i_t), output gate (o_t) and forget gate (f_t). Respectively, the activation of the input gate multiplies the input to the cells, the output gate multiples the output to the net, and the forget gate multiplies the previous cell values. Figure 2 shows the detail structure of the memory block.

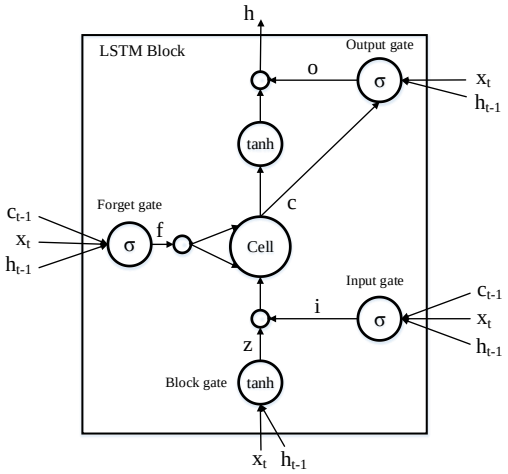

Figure 2 LSTM memory block

Consider x_i is the feature vector of the word, then the sequence data is denoted as $x_1, x_2, \ldots, x_i, x_m$, where m is the length of the sentence. Let h_{t-1} and c_{t-1} be the previous hidden and cell state of LSTM respectively. Thus, the computation of h_t and c_t would be:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right)$$

$$z_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_{c)}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot z_t$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$

$$h_t = o_t \cdot \tanh(c_t)$$

Here $W_{(.)}$ is the learning parameters for LSTM, o is the element-wise product, σ is the sigmoid activation function and b is the bias term.

As we employ the Bi-LSTM models, there would be two encoding results for the representation of the sequence data, h_t^f and h_t^b, which are produced by the forward LSTM layer and backward LSTM layer respectively. In this work, we add these two encoding results with the following operation:

$$z_t = h_t^f + h_t^b$$

C. Pooling layer

In this layer, we apply the piece max pooling, instead of max pooling, to get the optimal features from the entire sequence data, since the former method performs better than the latter method on obtaining more available information. Specifically, the encoding data learnt from Bi-LSTM layer is divided into several pieces with equal length. Let $z_k^1, z_k^2, \dots, z_k^t, z_k^l$ be the independent piece and $< v_t^1, v_t^2, \dots, v_t^d >_k$ be the vector of z_t^k, where k is the identifier of the piece, l is the length of the piece and d is the dimension. Then the result of piece max pooling would be:

$$z_k = < \max(v_t^1), \max(v_t^2), \dots, \max(v_t^d) >_k$$
$$z = z_1 + z_2 + z_k + \cdots + z_n$$

Where $\max(.)$ is to get the maximum value of each dimension wise. Subsequently, basing on the segmentation in the feature layer, we concatenate all the results as follow:

$$Z = z^{Ch_1} \oplus z^{Ch_2} \oplus z^{Ch_3}$$

D. Softmax layer

A softmax operation with dropout is set in this layer to give normalized probability score for each class. We use tanh as the activation function. The equations are given as follows:

$$h_s = \tanh(z)$$
$$p(y|x) = \text{Softmax}(W^S h^s + b^s)$$

Where W is the softmax matrix and b is the bias term.

E. Model training

We utilized the *word2vec* tool to map each word into a numeric vector for word embedding, and the position features are mapped into a vector with 10 binary components. In addition, the weights and biases in our model are update by backpropagation through time. Specifically, we choose the cross entropy loss function and Adam's technique (13) with gradient clipping, parameter averaging and L2-regularization to train our model.

III. Experimental settings

1. In the CHEMPROT track, and to focus mainly on a subset of key relevant relation types, all the annotated CHEMPROT relations (CPRs) were grouped into 10 semantically related classes that do share some under-lying biological properties. Those groups are labeled as [CPR:1, CPR:2, … CPR:10] and a detailed description is shown in Table 1.

Table 1 The description of CHEMPROT relations

Group	Eval.	CHEMPROT relations belonging to this group
CPR:1	N	PART_OF
CPR:2	N	REGULATOR|DIRECT_REGULATOR|INDIRECT_REGULATOR
CPR:3	Y	UPREGULATOR|ACTICATOR|INDIRECT_UPREGULATOR
CPR:4	Y	DOWNREGULATOR|INHIBITOR|INDIRECT_DOWNREGULATOR
CPR:5	Y	AGONIST|AGONIST-ACTIVATOR|AGONIST-INHIBITOR
CPR:6	Y	ANTAGONIST
CPR:7	N	MODULATOR|MODULATOR-ACTIVATOR|MODULATOR-INHIBITOR
CPR:8	N	COFACTOR
CPR:9	Y	SUBSTRATE|PRODUCT_OF|SUBSTRATE_PRODUCT_OF
CPR:10	N	NOT

2. To keep the generalizability of our model, two entities in a pair are respectively replaced with "*ENTITY_1*" and "*ENTITY _2*". For instance, the CPI candidates in the sentence " The activities of UGTs 1A3, 1A8, 1A9, 2B4 and 2B7 were low, whereas *UGT1A1* and *UGT2B17* exhibited no *HFC* glucuronidation activity" are blinded as shown in Table 2.

Table 2 An example of entity blinding

CPI candidate	Entities blinding
(*HFC, UGT1A1*)	The activities of UGTs 1A3, 1A8, 1A9, 2B4 and 2B7 were low, whereas ***ENTITY_2*** and UGT2B17 exhibited no ***ENTITY_1*** glucuronidation activity.
(*HFC, UGT2B17*)	The activities of UGTs 1A3, 1A8, 1A9, 2B4 and 2B7 were low, whereas UGT1A1 and ***ENTITY_2*** exhibited no ***ENTITY_1*** glucuronidation activity.

3. Our solution is built based on Tensorflow[1] package using Python. Table 3 lists the hyper parameters used in the experiments.

Table 3 The hyper parameters of our model

Parameter	Description	Value
dw	Dimension of word embedding	100
dp	Dimension of position embedding	10
num	The number of hidden units	200
ρ	The ratio of dropout	0.7
l_2	The L2 regularization	0.001
l_a	The learning rate of Adam optimizer	0.001

4. We evaluated our model on the sample set with 339 instances. Our model outperforms the baseline in terms of recall and F-score, as listed in Table 4. We manually inspected a number of wrongly classified instances and analyzed several examples in Table 5. It turns out that long and complex sentences, especially those with clauses, are the ones that are prone to misclassification. This can be attributed to the limitation of bi-directional long short term-memory network on learning syntactic information from the extreme long and complex text in practice, although it can process the long

[1] https://www.tensorflow.org

sequential data in theory. More research needs to be done in order to address this kind of phenomenon.

Table 4 Performance comparison

	Precision	Recall	F-score
Our model	73.73	**66.95**	**70.16**
Baseline model	**85.21**	50.63	63.52

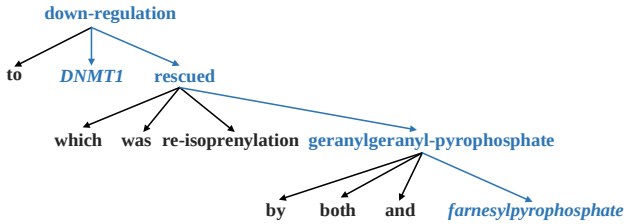

Figure 2 Part of dependency tree of instance S1

Table 5 Some examples of the classification errors

No.	Instance	Label	Prediction
S1	Further we found stimulation of FAS-expression as a result of epigenetic DNA demethylation that was due to down-regulation of ***DNMT1***, which was rescued by re-isoprenylation by both *geranylgeranyl-pyrophosphate* and ***farnesylpyrophosphate***.	CPR:3	CPR:4
S2	Our study shows that human TRPA1 is a target for apomorphine, suggesting that an activation of ***TRPA1*** might contribute to adverse side effects such as nausea and painful injections, which can occur during treatment with ***apomorphine***.	CPR:3	CPR:4
S3	Treatment of cells with BCNU to inhibit glutathione reductase (GR) enhanced the CpG-induced intracellular oxidation and decreased the ***GSH***/GSSG, with increased activation of NF-kappaB and a doubling in the CpG-induced production of ***IL-6*** and TNF-alpha.	CPR:4	CPR:3

ACKNOWLEDGMENT

This work is funded by the National Key Research and Development Program of China 2016YFB0200401.

REFERENCES

1. Segura Bedmar I. Application of information extraction techniques to pharmacological domain: extracting drug-drug interactions[J]. 2010.
2. Corney, D.P., Buxton, B.F., Langdon, W.B. and Jones, D.T. (2004) 'BioRAT: extracting biological information from full-length papers', Bioinformatics, Vol. 20, No. 17, pp.3206–3213.
3. Garc á-Blasco S, Danger R, Rosso P. Drug-Drug interaction detection: A new approach based on maximal frequent sequences[J]. Procesamiento del Lenguaje Natural, 2010, 45: 263-266.
4. Bj örne J, Kaewphan S, Salakoski T. UTurku: drug named entity recognition and drug-drug interaction extraction using SVM classification and do-main knowledge[C]//Second Joint Conference on Lexical and Computational Semantics (* SEM). 2013, 2: 651-659.
5. Liu S, Tang B, Chen Q, et al. Drug-Drug Interaction Extraction via Convolutional Neural Networks[J]. Computational & Mathematical Methods in Medicine, 2016, 2016:1-8.
6. Xiao, J., Su, J., Zhou, G.D. and Tan, C. (2005) 'Protein–protein interaction extraction: a supervised learning approach', Proceedings of the Symposium on Semantic Mining in Biomedicine, European Bioinformatics Institute, Hinxton, UK, pp.51–59.
7. Bunescu R, Mooney R. Subsequence kernels for relation extraction. In: Proceedings of NIPS'2005. p. 171–8.
8. Chowdhury FM, Lavelli A, Moschitti A. A study on dependency tree kernels for automatic extraction of protein–protein interaction. In: Proceedings of BioNLP'2011. p. 124–33.
9. Hua, Lei, Quan, Chanqin. A Shortest Dependency Path Based Convolutional Neural Network for Protein-Protein Relation Extraction. BioMed Research International, 2016: 1-9
10. Hsieh, Yu-Lun & Chang, Yung-Chun & Chang, Nai-Wen & Hsu, Wen-Lian. (2017). Identifying Protein-protein Interactions in Biomedical Literature using Recurrent Neural Networks with Long Short-Term Memory. The 8th International Joint Conference on Natural Language Processing.2017.
11. Bobic T, Fluck J, Hofmannapitius M. SCAI: Extracting drug-drug interactions using a rich feature vector[J]. Relation Extraction, 2013.
12. Bj örne J, Kaewphan S, Salakoski T. UTurku: drug named entity recognition and drug-drug interaction extraction using SVM classification and domain knowledge[C]//Second Joint Conference on Lexical and Computational Semantics (* SEM). 2013, 2: 651-659.
13. Kingma D, Ba J. Adam: A method for stochastic optimization[J]. arXiv pre-print arXiv:1412.6980, 2014.