Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Subject Section

Position-Aware Deep Multi-Task Learning for Drug-Drug Interaction Extraction

Lei Miao[1], Deyu Zhou[1,*] and Yulan He[2]

[1]School of Computer Science and Engineering, Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing, China.
[2]School of Engineering and Applied Science, Aston University, Birmingham, UK

*To whom correspondence should be addressed.

Associate Editor:

Received on ; revised on ; accepted on

Abstract

Motivation: A drug-drug interaction (DDI) is a situation in which a drug affects the activity of another drug synergistically or antagonistically when being administered together. The information of DDIs is crucial for healthcare professionals to prevent adverse drug events. Although some known DDIs can be found in purposely-built databases such as DrugBank, most information is still buried in scientific publications. Therefore, automatically extracting DDIs from biomedical texts is sorely needed. In this paper, we propose a novel position-aware deep multi-task learning approach for extracting DDIs from biomedical texts. In particular, sentences are represented as a sequence of word embeddings. An attention-based bidirectional long short-term memory (BiLSTM) network is used to encode each sentence. The relative position information of words with the target drugs in text is combined with the hidden states of BiLSTM to generate the position-aware attention weights. Moreover, the tasks of predicting whether or not two drugs interact with each other and further distinguishing the types of interactions are learned jointly in multi-task learning framework.
Results: The proposed approach has been evaluated on the DDIExtraction challenge 2013 corpus and the results show that with the position-aware attention only, our proposed approach outperforms the state-of-the-art method by 1.16% for binary DDI classification, and with both position-aware attention and multi-task learning, our approach achieves a micro F-score of 73.14% on interaction type identification, outperforming the state-of-the-art approach by 1.66%, which demonstrates the effectiveness of the proposed approach.
Availability: The source code of the proposed approach and the dataset used are freely available for non-commercial purposes at `http://cse.seu.edu.cn/people/zhoudeyu/`.
Contact: d.zhou@seu.edu.cn

1 Introduction

A drug-drug interaction (DDI) is a situation in which a drug affects the activity of another drug synergistically or antagonistically when being administered together. Concomitant medications might alter drug transportation abruptly in individuals who have previously taken a particular dose of a drug. Such an abrupt alteration might change the known safety and efficacy of a drug. For example, *terfenadine* was a common antihistamine intended to block the effects of an allergic rhinitis. Unfortunately several people who took *terfenadine* concomitantly with *ketoconazole*, an antifungal, suffered cardiac problems which often lead to death (PK *et al.*, 1993). Therefore, it is crucial to extract the information about DDIs. Although some known DDIs can be found in drug-related databases such as DrugBank [1], most information is still buried in scientific articles. Automatic DDI extraction, aiming to automatically discover DDIs from text with high efficiency and accuracy, is becoming an increasingly well understood alternative to manual DDI discovery. Without automated DDI extraction tools, it is hard for doctors, pharmacists and researchers to keep up with the most recent discoveries described in biomedical literature.

To tackle the DDI extraction problem, several evaluation tasks, such as DDIExtraction 2011 (Segura Bedmar *et al.*, 2011) and DDIExtraction

[1] `http://www.drugbank.ca/`

2013 (Segura Bedmar *et al.*, 2013) shared tasks, have been proposed in recent years to provide common benchmarking datasets for the evaluation of DDI detection from biomedical text. An example of a sentence and its corresponding DDI annotations selected from DDIExtraction 2013 is presented in Table 1. The sentence contains three drug entities: "*neomycin sulfate*", "*coumarin*" and "*anticoagulants*". The DDI with the interaction type "*effect*" exists between *neomycin sulfate* and *coumarin*, and also, *neomycin sulfate* and *anticoagulants*. However, for the drug pair *coumarin* and *anticoagulants*, there is no interaction between them. As such, its interaction type is annotated as "*other*". There are two DDI detection tasks defined, first, given a drug pair, determine whether there exists a DDI between them; second, determine the type of DDI interaction. The former task is essentially a binary classification problem while the latter is a multi-class classification problem since there are more than two interaction types.

Table 1. An example of one sentence and its corresponding DDI annotations selected from DDIExtraction 2013.

Sentence	Oral **neomycin sulfate** may enhance the effect of **coumarin** in **anticoagulants** by decreasing vitamin K availability.
Annotation	Drug1: neomycin sulfate, Drug2: coumarin, Type: effect Drug1: neomycin sulfate, Drug2: anticoagulants, Type: effect Drug1: coumarin, Drug2: anticoagulants, Type: other

Early approaches to automatic extraction of DDIs are mostly based on hand-crafted rules due to the lack of annotated datasets (Segura-Bedmar *et al.*, 2011a). With the introduction of DDIExtraction challenges in 2011 (Segura Bedmar *et al.*, 2011) and 2013 (Segura Bedmar *et al.*, 2013) and the availability of annotated DDI datasets, more and more machine learning based methods have been proposed (Bui *et al.*, 2014). These approaches typically rely on a set of carefully designed features to train supervised classifiers such as support vector machine (SVM). The results of DDI extraction largely depend on the quality of the features used, as evidenced by the submitted systems to the DDIExtraction 2013 challenge. To avoid the tedious process for feature design, in recent years, deep learning techniques have been proposed to automatically learn feature representations from abundant unannotated data (Bengio *et al.*, 2013). Different neural network based methods, such as Convolutional Neural Networks (CNNs) (Liu *et al.*, 2016b) and Recurrent Neural Networks (RNNs) (Sahu and Anand, 2017), have been proposed to automatically extract feature vectors from sentences for DDI extraction.

Nevertheless, we argue that the position that a drug occurs in a sentence could be important for DDI extraction since drug pairs occur in different positions could capture syntactic information to a certain extend and hence gives indications of DDIs. Also, binary (classify whether DDI exists within the given drugs) or not) and multi-class (classify the interaction of a drug pair into one of the DDI types) DDI classifications are related tasks. Hence, by learning the two tasks jointly, we can capture the shared features effectively which might benefit to each other. In this paper, we propose a novel position-aware deep multi-task learning approach which is built upon bidirectional long short-term memory networks, called PM-BLSTM, for extracting DDIs from biomedical texts. Our contributions are summarized below: (1) We incorporate the position-aware attention mechanism by combining position embeddings with the hidden states of BiLSTM to generate the position-aware attention weights. Moreover, the position embedding is employed repeatedly when generating the attention weights to make the attention mechanism more flexible and potent; (2) We propose a multi-task learning framework to tackle jointly the tasks of predicting whether or not two drugs interact with each other and further distinguishing the types of interactions. Jointly learning both tasks allows the capture of shared features more effectively which could benefit both tasks; (3) The proposed approach has been evaluated on the DDIExtraction challenge 2013 corpus and the results show that with the position-aware attention only, our proposed approach outperforms the state-of-the-art method by 1.16% for binary DDI classification, and with both position-aware attention and multi-task learning, our approach achieves a micro F-score of 73.14% on interaction type identification, outperforming the state-of-the-art approach by 1.66%, which demonstrates the effectiveness of the proposed approach.

The rest of the paper is organized as follows. Section 2 surveys existing approaches for DDI extraction. Section 3 describes the proposed method, which consists of four components: the embedding layer, the BiLSTM layer, the position-aware attention layer and the multi-task output layer. Section 4 presents experimental setup and results. Finally, Section 5 concludes the paper and outlines future research directions.

2 Related Work

Most existing approaches to DDI extraction are based on machine learning. In order to predict the relation between a given pair of drugs, classifiers are typically trained on lexical, syntactic and semantic features extracted from manually labelled corpora. Based on the way of feature construction, approaches can be roughly divided into two categories, feature-based and kernel-based methods (Bui *et al.*, 2014). Feature-based approaches focus on finding potentially discriminative features to represent data characteristics. Apart from the basic bag-of-words features, researchers have explored the use of various types of features including context features (Segura-Bedmar *et al.*, 2011b), a combination of lexical, semantic and domain features (He *et al.*, 2013), and heterogeneous features consisting of lexical, syntactic, semantic and negation features derived from parse trees (Chowdhury and Lavelli, 2013a). Kernel-based approaches employ different kernels to calculate the similarity between two instances by exploiting the structural representations of data instances such as syntactic parse trees or dependency graphs (Tikk *et al.*, 2013). In the past DDIextraction challenges, the most commonly used kernels are all-paths graph kernel (Airola *et al.*, 2008), shallow linguistic kernel (Giuliano *et al.*, 2006) and path-enclose tree kernel (Moschitti, 2004). It is also possible to combine multiple kernels in order to compensate the weakness of each individual kernel. For example, in the work of Faisal *et al.* (2013), three different kernels were combined to form a hybrid kernel which gives a better performance compared to those using a single kernel. However, these approaches typically rely on feature engineering to generate a list of discriminative features for training supervised classifiers. As observed in DDIExtraction 2013 challenge, different approaches adopted different feature engineering techniques and there is no standard way in generating features. Moreover, features often need to be redesigned when previously developed systems are adapted for the task of DDI extraction. For example, UTurku, a system originally developed for biomedical event extraction was adapted for DDI extraction by redesigning the features specifically for DDI extraction (Jari Björne and Salakoski, 2013).

In recent years, deep learning techniques have been proposed to automatically learn feature representations from abundant unannotated data (Bengio *et al.*, 2013). Features for DDI extractor can be learned automatically using deep neural networks without expensive manual feature engineering. Based on the structure of neural network, these methods can be roughly classified into two categories: CNN based models and RNN based models. Liu *et al.* (2016b) attempted to use CNN for DDIs extraction. They adopted a shallow CNN and combined word embeddings with position embeddings in the CNN model. Liu *et al.* (2016a) introduced the structure information of sentences with dependency-based CNN for

DDI extraction. Quan *et al.* (2016) incorporated semantic information of multiple word embeddings with multichannel CNNs. Zhao *et al.* (2016) combined CNN with some traditional features. Apart from CNNs, RNNs have also been used for biomedical relation classification. Sahu and Anand (2017) used LSTMs for DDI extraction and Yi *et al.* (2017) extracted DDI at the corpus level via gated recurrent unit networks (GRUs). In both approaches, the attention mechanism was incorporated.

3 Method

In this section, we present the proposed position-aware multi-task deep learning method built on bidirectional LSTMs (PM-BLSTM), which is illustrated in Figure 3, for DDI extraction. It can be observed that PM-BLSTM is based on a BiLSTM by incorporating position-aware attentions. Different from previous attention-based LSTM based approaches (Zhou *et al.*, 2016), PM-BLSTM utilizes the position information to enhance the effectiveness of the attention mechanism. Moreover, the tasks of predicting whether or not two drugs interact with each other and further distinguishing the types of interactions are tackled jointly in output layer. The proposed approach consists of four main components: the embedding layer, the BiLSTM layer, the position-aware attention layer and the multi-task output layer. In the following, we first discuss how we pre-process the data and then describe each of the four components of PM-BLSTM in details.

3.1 Preprocessing

As the proposed approach is to predict whether there exists an interaction between two drugs and identify the interaction type, the sentence with more than two drug entities needs to be processed to make sure only one drug pair remains in each instance. For the sentence containing more than two drug entities, C_n^2 instances are generated. Following the previous method (Kim *et al.*, 2015), we replace the two target drug entities with symbols "DRUG1" and "DRUG2" respectively, and represent other drug entities as "DRUG0". The drug pairs with interactions are considered as positive training instances and the others are considered as negative training instances. However, such preprocessing might generate some redundant and ambiguous training instances. For example, for the sentence "$drug_1$/$drug_2$: $drug_3$ inhibits the enzymatic oxidation of $drug_4$ and $drug_5$ to $drug_6$.", it contains two positive DDIs $\langle drug_3, drug_4\rangle$ and $\langle drug_3, drug_5\rangle$ while the other 13 drug pairs such as $\langle drug_1, drug_2\rangle, \langle drug_1, drug_3\rangle, \langle drug_1, drug_4\rangle$ are all negative instances. The number of negative instances is significantly more than that of positive instances. Therefore, we defined two rules below for filtering the generated noisy training instances:

- Rule 1: Instances with two target drugs referring to the same drug should be removed.
- Rule 2: Instances with two target drugs being in coordinate position should be removed.

3.2 Embedding Layer

The input to the embedding layer is DDI instances. Given a sentence $S = \{w_1, w_2, ..., w_T\}$ in which w_u ="DRUG1" and w_v ="DRUG2", two position measures p_{i_1} and p_{i_2} are defined for each word w_i. Here, $P_{i_1} = i - u$ and $P_{i_2} = i - v$, representing the relative distance between w_i and the target drug w_u and w_v respectively. In this layer, each word w_i is represented by a d_w-dimensional vector e_{w_i}, by looking up a word embedding dictionary which can be initialized randomly or uses the pre-trained vectors. Position measures are embedded in the d_p-dimensional space which are initialized randomly. The whole process can be described as follows:

$$e_{w_i} = LT_w(w_i), \quad e_{p_{i_1}} = LT_p(p_{i_1}), \quad e_{p_{i_2}} = LT_p(p_{i_2}), \tag{1}$$

where LT denotes a look-up operation. After that, the word embedding e_{w_i} and the two position embeddings $e_{p_{i_1}}$ and $e_{p_{i_2}}$ are concatenated to generate the final embedding x_i for the word w_i. Hence, $x_i \in \mathbb{R}^{(d_w+2d_p)}$:

$$x_i = e_{w_i} \oplus e_{p_{i_1}} \oplus e_{p_{i_2}}, \tag{2}$$

where $\oplus$ denotes a concatenation operation. Finally, a sentence with T words is represented by $\{x_1, x_2, ...x_T\}$, which forms the input to the BiLSTM Layer.

3.3 Bidirectional LSTM Layer

Recurrent neural network (RNN) is a powerful model for processing serialized input with arbitrary length. As a special RNN structure, LSTM was proposed to address the problem of exploding or vanishing gradients (Hochreiter and Schmidhuber, 1997). In general, LSTM possesses a memory cell which is able to store the previous information over long periods of time. An adaptive gating mechanism is utilized for restricting the previous state and the current input information.

The LSTM unit at each time step shares the same structure and parameters. For example, at time t, an LSTM unit receives the previous hidden state h_{t-1} and the current input vector x_t. Based on h_{t-1} and x_t, the input gate i_t, the forget gate f_t and the output gate o_t are calculated accordingly. The memory cell c_t absorbs both the current and previous information restricted by i_t, f_t. The complete neural network can be calculated by the following formulae:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \tag{3}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f), \tag{4}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{5}$$

$$g_t = tanh(W_c x_t + U_c h_{t-1} + b_c), \tag{6}$$

$$c_t = i_t \odot g_t + f_t \odot c_{t-1}, \tag{7}$$

$$h_t = o_t \odot tanh(c_t), \tag{8}$$

where σ is the logistic sigmoid function and $\odot$ denotes element wise multiplication. All of the vectors in the Left-Hand Side of the equations (3) − (8) are in $\mathbb{R}^d$.

To consider both directional information of an input sentence, we adopt the bidirectional LSTM (BiLSTM) network with two LSTM layers to process the sequence in forward and backward directions respectively. The hidden states of the ith step are are concatenated as $h_i = [\overrightarrow{h_i} \oplus \overleftarrow{h_i}]$. Thus, we get a $2d$-dimensional vector at each time step.

3.4 Position-Aware Attention Layer

Attention-based models have demonstrated success in a wide range of NLP tasks (Zhou *et al.*, 2016; Bahdanau *et al.*, 2014). The basic idea of the attention mechanism is to assign a weight to each hidden unit in the lower-level of the neural network when computing an upper-level representation. However, for DDI extraction, the biomedical sentences are always long-winded and often contain more than one drug pair. Therefore, we propose a position-aware attention mechanism which considers not only the semantic information of words but also the global position information.

Let $H \in \mathbb{R}^{2d \times T}$ be a matrix consisting of the hidden states $[h_1, h_2, ..., h_T]$ produced by the BiLSTM layer. Let $E \in \mathbb{R}^{2d_p \times T}$ be a matrix consisting of the position vectors $[e_1, e_2, ..., e_T]$, where

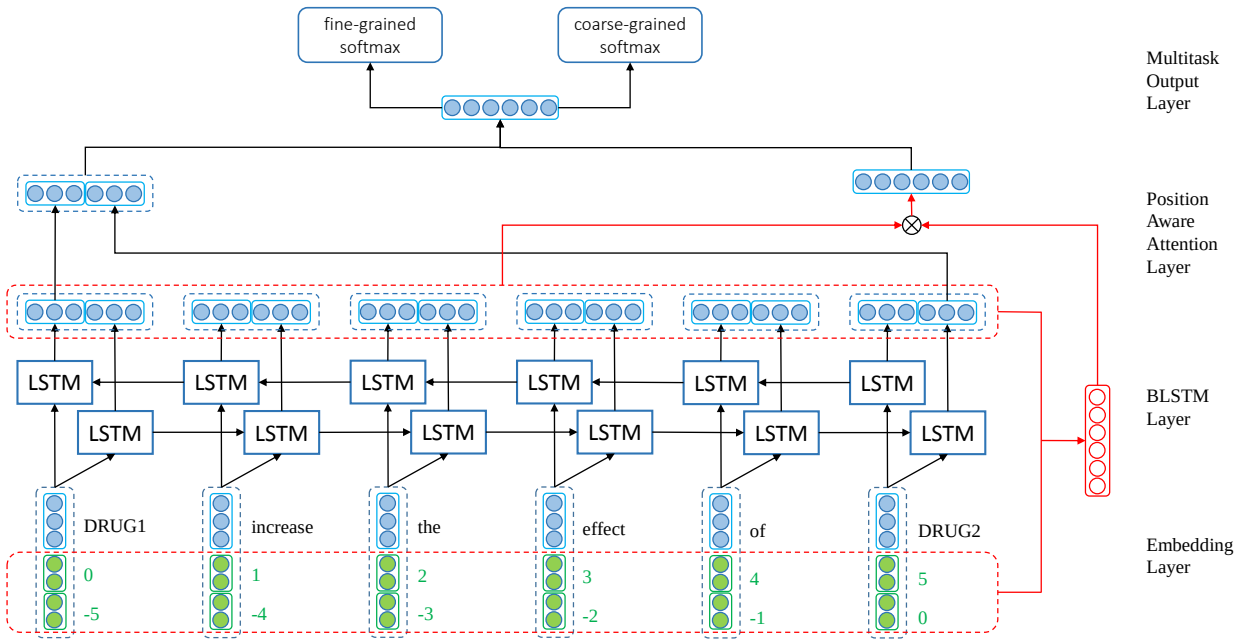

Fig. 1. The architecture of the position-aware multi-task bidirectional LSTM (PM-BLSTM).

$e_i = [e_{p_{i_1}}{}^\intercal, e_{p_{i_2}}{}^\intercal]^\intercal$. The attention weight α and the weighted hidden representation r are calculated based on the following formulae:

$$M = tanh(\begin{bmatrix} W_h H \\ W_p E \end{bmatrix}), \quad (9)$$

$$\alpha = softmax(w^T M), \quad (10)$$

$$r = H\alpha^T, \quad (11)$$

where $M \in \mathbb{R}^{2d+2d_p \times T}$, $\alpha \in \mathbb{R}^T$, $r \in \mathbb{R}^{2d}$. $W_h \in \mathbb{R}^{2d \times 2d}$, $W_p \in \mathbb{R}^{2d_p \times 2d_p}$ and $w \in \mathbb{R}^{2d+2d_p}$ are all transformation matrices need to be learned.

In order to increase the diversity, the last hidden state of BiLSTM is also used. Let $\overrightarrow{h_T} \in \mathbb{R}^d$ be the last hidden state of the forward LSTM, and $\overleftarrow{h_1} \in \mathbb{R}^d$ be the last hidden state of the backward LSTM corresponding to the first word. $h_l \in \mathbb{R}^{2d}$ is the concatenation of h_T and h_1 which represents the output of the BiLSTM layer. The final sentence representation is calculated by:

$$h_l = [\overrightarrow{h_T} \oplus \overleftarrow{h_1}], \quad (12)$$

$$h^* = tanh(W_p r + W_x h_l), \quad (13)$$

where $h^* \in \mathbb{R}^{2d}$, $W_p \in \mathbb{R}^{2d \times 2d}$ and $W_x \in \mathbb{R}^{2d \times 2d}$ are both affine transformation matrices.

3.5 Multi-Task Learning

In DDI extraction, there are two subtasks. **Task-1** is a binary classification problem which aims to predict whether or not two drugs interact with each other. **Task-2** is a multi-class or more fine-grained classification problem, which aims to further distinguish the types of interactions. In the DDI extraction 2013 challenge, four types of DDIs are defined including "*advice*", "*effect*", "*mechanism*" and "*intact*". Previously, the two subtasks are learned separately. However, the two subtasks are closely related. As such, we explore the use of multi-task learning here. Multi-task learning is an approach to improve generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It achieves this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better (Caruana, 1997).

In the multi-task output layer, two $softmax$ based classifiers, y_c and y_f, are used for coarse-grained (or binary) classification and fine-grained (or multi-class) classification tasks respectively. The dimensions of y_c and y_f correspond to the number of the classes in the two classification tasks.

$$y_c = softmax(W_{sc} h^* + b_{sc}), \quad (14)$$

$$y_f = softmax(W_{sf} h^* + b_{sf}), \quad (15)$$

where $W_{sc}, b_{sc}, W_{sf}, b_{sf}$ are all the parameters of $softmax$ functions.

The objective function is the negative log-likelihood of predicted results y:

$$loss = -\sum_{i=1}^{m} t^i log(y^i) + \lambda \|\theta\|^2, \quad (16)$$

where $\mathbf{t} \in \mathcal{R}^m$ is one-hot vector (m is the number of classes), λ is an $L2$ regularization parameter and θ is the parameter set.

4 Experiments

In this section, we present the experimental setup and results for the evaluation of the effectiveness of the proposed approach.

4.1 Experimental Setup

We use the datasets provided by the DDI extraction 2013 challenge to evaluate the proposed approach. There are two datasets constructed. One is from the DrugBank database (DB-2013) and the other is from MedLine

abstracts (ML-2013). DB-2013 and ML-2013 are combined together and the training/testing split follows the same experimental setup in most neural network based approaches (Liu *et al.*, 2016b). The statistics of the datasets are shown in Table 2. The pre-processing rules mentioned in Section 3 are applied on the training and testing data to remove ambiguous or misleading instances. It can be observed from the lower part of Table 2 that more than 35% of negative instances are filtered.

Table 2. Statistics of the datasets used in the experiments.

	Training			Testing		
	DB-2013	ML-2013	ALL	DB-2013	ML-2013	ALL
Negative	22118	1547	23665	4367	345	4712
Positive	3788	232	4020	884	95	979
Effect	1535	152	1687	298	62	360
Mechanism	1257	62	1319	278	24	302
Advice	818	8	826	214	7	221
Intact	178	10	188	94	2	96
After preprocessing						
Negative	14208	1181	15389	2732	243	2975
Positive	3750	231	3981	884	91	979
Effect	1510	152	1662	298	61	359
Mechanism	1250	62	1312	278	21	299
Advice	813	7	820	214	7	221
Intact	177	10	187	94	2	96

In our experiments, position embeddings are initialized randomly. The dimension of position embeddings is set to 10. Word embeddings are pre-trained using unlabeled biomedical texts we crawled from PubMed using Word2vec (Mikolov *et al.*, 2013). The dimension of word embeddings is set to 300. The dimension of the BiLSTM hidden layer is set to 150. All models are trained with a batch size of 5 instances and the neural networks are optimized with $AdaMax$. To overcome the overfitting problem, the $L2$ regularization weight is set to 10^{-4} in the BiLSTM layer. Dropout is used in the embedding layer, the BiLSTM layer and the output layer.

To evaluate the performance of the propose approach, we use Precision (*P*), Recall (*R*) and F-score (*F*) which are commonly used for the evaluation of classification results. For **Task-2**, we calculate the overall micro precision ($pmicro\text{-}P$), recall ($micro\text{-}R$) and F-score ($micro\text{-}F$) defined as follows, which are the standard evaluation metrics used in the DDIExtraction 2013 challenge. $micro\text{-}P = \overline{TP}/(\overline{TP} + \overline{FP})$, $micro\text{-}R = \overline{TP}/(\overline{TP} + \overline{FN})$, $micro\text{-}F = 2 \times micro\text{-}P \times micro\text{-}R/(micro\text{-}P + micro\text{-}R)$, where $-$ denotes the average value calculated across different classes, TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives.

4.2 Overall Comparison

We compare our approach with the baseline models in two categories: traditional methods and neural network based approaches. Traditional methods utilize manually designed features to train supervised classifiers for DDI extraction:

- UTurku (Björne *et al.*, 2013) was adapted from the Turku Event Extraction System (TEES), which used features from dependency parsing and domain dependent resources.
- WBI (Thomas *et al.*, 2013) combined features of a number of DDI approaches.
- FBK-irst (Chowdhury and Lavelli, 2013b) used linear features, path-enclosed tree kernels and shallow linguistic features.
- Kim (Kim *et al.*, 2015) used contextual, lexical, semantic and tree structured features.

Neural network based approaches learn feature representations of instances automatically based on different neural network structures.

- CNN (Liu *et al.*, 2016b) is a shallow convolutional neural network with both word and position embeddings.
- SCNN (Zhao *et al.*, 2016) is an CNN with manually designed features.
- MCNN (Quan *et al.*, 2016), semantic information is introduced as multichannel word embedding for CNN.
- DCNN (Liu *et al.*, 2016a), a dependency-based CNN for DDI extraction.
- Joint AB-LSTM (Sahu and Anand, 2017), the outputs of a classical LSTM and an attention-based LSTM is jointly learned.

As our approach consists of two key components, position-aware attention mechanism and multi-task learning, we have also implemented two variants of our model, one with only position-aware attention mechanism (called P-BLSTM) and another with only multi-task learning (called M-BLSTM).

Table 3 shows the results of the proposed approach in comparison with the baselines for Task-1 and Task-2. It can be observed that for Task-1, the feature-based approach FBK-irst gives balanced precision and recall values and achieves the F-score of 80%. The recent NN-based approach Joint AB-LSTM improves upon the precision significantly but with worse recall in comparison to FBK-irst, which gives marginally better F-score. Our proposed PM-BLSTM performs slightly better in recall, but worse in precision compared to Joint AB-LSTM, and gives the F-score which is 1.24% lower. However, a variant of our model P-BLSTM without multi-task learning outperforms PM-BLSTM in recall and achieves the best F-score of 81.48% overall, outperforming the state-of-the-art approach Joint AB-LSTM by 1.16%.

For Task-2 which is a more difficult multi-class classification problem, we notice that NN-based approaches in general outperform feature-based approaches. We also observe that among all NN-based approaches, our PM-BLSTM achieves the best micro-recall value and comparable micro-precision value. Overall, PM-BLSTM outperforms the state-of-the-art approach, Joint AB-LSTM, by 1.66% in micro-F-score.

4.3 The Impact of Multi-Task Learning

To further investigate the effectiveness of incorporating multi-task learning, we compare P-BLSTM without multi-task learning with PM-BLSTM in more details in Table 4. It can be observed that the micro F-score of DDI extraction on Task 2 is improved by over 0.7% with multi-task learning. However, for Task 1, incorporating multi-task learning leads to slight degradation of F-score. One possible reason is that multi-task learning can get relevant inductive bias by sharing the related information of different tasks. For Task-2, the number of positive instances for each interaction type is much lower than the negative instances. Learning both Task-1 and Task-2 jointly allows the sharing of low-level features and hence alleviate the imbalanced data problem. However, the class imbalance problem is less severe for Task-1 compared to Task-2. Hence, leveraging features learned in Task-2 for classification in Task-2 does not give any performance gains. As such, we can conclude that multi-task learning helps in the more difficult multi-class classification problem (Task-2), but appears to be less effective for the binary classification Task-1.

4.4 The Impact of Position-Aware Attention

We also compare M-BLSTM with PM-BLSTM in more details in Table 5 to further investigate the effectiveness of incorporating position-aware attention. It can be observed that the position-aware attention mechanism appears to be very important as it improves the F-score for Task-1 by over 0.4% and the micro F-score For Task-2 by over 1.2%. We also observe

Table 3. Performance comparison of DDI extraction with other baselines.

Category	Methods	Task-1			Task-2		
		P	*R*	*F*	*micro-P*	*micro-R*	*micro-F*
Traditional Methods	UTurku	85.80	58.50	69.60	73.20	49.90	59.40
	WBI	80.10	72.20	75.90	64.20	57.90	60.90
	FBK-irst	79.40	**80.60**	80.00	64.60	65.60	65.10
	Kim	-	-	77.50	-	-	67.00
Neural Network	CNN	-	-	-	75.70	64.60	69.75
	SCNN	77.50	76.90	77.20	72.50	65.10	68.60
	MCCNN	-	-	-	75.99	65.25	70.21
	DCNN	-	-	-	**78.24**	64.66	70.81
	Joint AB-LSTM	**86.36**	75.07	80.32	73.41	69.66	71.48
Our approach	PM-BLSTM	82.90	75.59	79.08	75.80	70.67	**73.14**
	P-BLSTM	82.68	80.31	**81.48**	74.57	70.36	72.40
	M-BLSTM	83.78	74.15	78.67	71.90	**71.90**	71.90

Table 4. DDI extraction results with or without multi-task learning.

		P-BLSTM			PM-BLSTM		
		P	*R*	*F*	*P*	*R*	*F*
Task-1	Positive	82.68	80.31	**81.48**	82.90	75.59	79.08
Task-2	Effect	69.60	72.70	71.12	69.58	73.26	**71.37**
	Mechanism	76.92	66.89	71.56	80.93	69.57	**74.82**
	Advice	79.57	84.62	**82.02**	80.00	83.26	81.60
	Intact	76.00	39.58	**52.05**	77.27	35.42	48.57
	Overall (micro)	74.57	70.36	72.40	75.80	70.67	**73.14**

consistently better results for all interaction types. The results demonstrate the effectiveness of using position-aware attention for DDI extraction.

Table 5. DDI extraction results with or without position-aware attention.

		M-BLSTM			PM-BLSTM		
		P	*R*	*F*	*P*	*R*	*F*
Task-1	Positive	83.78	74.15	78.67	82.90	75.59	**79.08**
Task-2	Effect	67.43	73.82	70.48	69.58	73.26	**71.37**
	Mechanism	72.55	74.25	73.39	80.93	69.57	**74.82**
	Advice	79.20	81.00	80.09	80.00	83.26	**81.60**
	Intact	70.00	36.46	47.95	77.27	35.42	**48.57**
	Overall (micro)	71.90	71.90	71.90	75.80	70.67	**73.14**

An example of two instances are presented in Figure 2 to illustrate the effect of using position-aware attention comparing with the normal attention mechanism. Here, one word's attention weight is represented by the intensity of the red color. Darker color denotes higher attention weights while lighter color represents lower attention weights. It can be observed that the proposed position-aware attention can successfully locate the keywords in sentences which indicate the interaction on the specific drug-drug pairs. Compared with the normal attention mechanism, the proposed position-aware attention can identify the location of keywords more accurately by making full use of words' relative distance with the target drug entities. For example, in the first sentence, the relation between DRUG1 and DRUG2 is expressed in the first clause. The normal attention mechanism generates the largest attention weights on "*metabolism*" and the second largest weight on "*concentration*" in the second clause which is irrelevant. But the position-aware attention focuses correctly on the keyword "*reduces*" which is the key indicator of the DDI.

4.5 The Impact of Preprocessing

The ratio of positive to negative classes in the original training set is 1:5.9. The highly imbalance nature of the corpus makes it difficult to learn classifier for DDI extraction. The problem can be partially alleviated by filtering out some ambiguous negative instances before training. But the preprocessing method would also lose some potentially useful information in the filtered instances. Therefore, the filtering rules should strike a balance between the two factors. Table 6 shows the performance of PM-BLSTM with or without negative instance filtering. It can be observed that preprocessing gives better results on both Task-1 and Task-2. As such, negative instance filtering appears to be useful for DDI classifier learning.

Table 6. DDI extraction results with or without preprocessing.

	Task-1			Task-2		
	P	R	F	micro-P	micro-R	micro-F
without preprocessing	82.41	75.82	78.98	72.98	70.29	71.61
with preprocessing	82.90	75.59	79.08	75.80	70.67	73.14

4.6 Error Analysis

We conduct error analysis to gain a better insight of our proposed approach. We have identified three factors which might lead to classification errors. (1) data imbalance: as shown in Table 4, the proposed approach achieves the worst performance on the interaction type "*intact*" as there are only 187 training instances in such a category. Instances with "*intact*" are often misclassified as "*effect*". (2) complex sentence structures: long sentences and implicit descriptions always make classification hard. For example, for the sentence "*...therefore, do not administer DRUG1 with DRUG0 or other agents that may interfere with enterohepatic recirculation or drugs that may bind bile acids, for example, bile acid sequestrates or oral DRUG2, because of the potential to reduce the efficacy of DRUG0.*", the relation

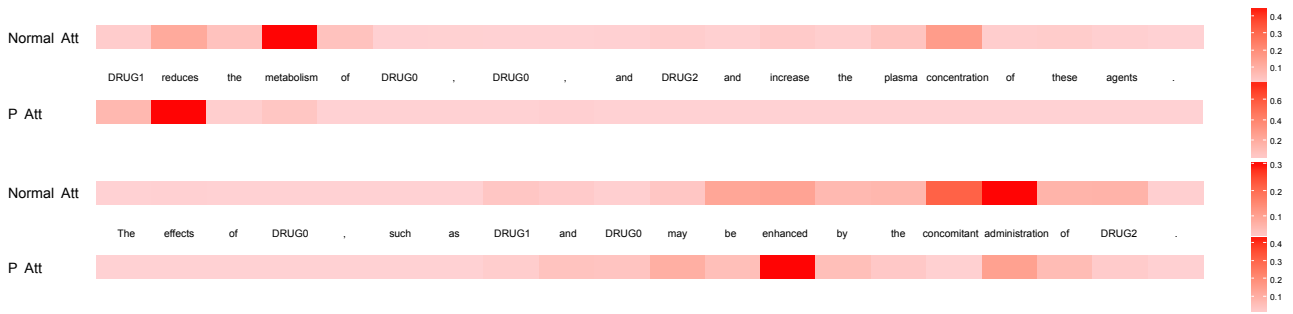

Fig. 2. Comparison of normal attention (normal Att) and the proposed position-aware attention (P Att).

between DRUG1 and DRUG2 is described implicitly. Here, "DRUG2" is the appositive of "*other agents*" which has been directly expressed to have an interaction with "DRUG1". But there is an attributive clause between "*other agents*" and "DRUG2". which makes the detection of DDI between DRUG1 and DRUG2 difficult. (3) lack of training data: it is well known that deep neural network based methods need abundant training instances. The data in the DDIExtraction 2013 challenge is not large enough which makes it difficult to improve the performance of NN-based approaches.

5 Conclusion

In this paper, we have proposed a novel multi-task recurrent neural network architecture with position-aware attentions (PM-LSTM) for DDI extraction. To improve the efficiency of the attention mechanism, PM-LSTM utilizes an additional position embedding to generate the attention weights. Besides, the model takes the advantage of multi-task learning by predicting whether or not two drugs interact with each other and further distinguishing the types of interactions jointly. Experimental results on DDIExtraction 2013 corpus show that with the position-aware attention only, our proposed approach outperforms the state-of-the-art method by 1.16% for binary DDI classification, and with both position-aware attention and multi-task learning, our approach achieves a micro F-score of 73.14% on distinguishing the types of interactions, outperforming the state-of-the-art approach by 1.66%. In the future, we will explore other structures of neural networks for multi-task learning so as to improve the DDI extraction performance of both tasks simultaneously.

Acknowledgements

This work was funded by the National Natural Science Foundation of China (61528302, 61772132), the Natural Science Foundation of Jiangsu Province of China (BK20161430) and the Collaborative Innovation Center of Wireless Communications Technology.

References

Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., and Salakoski, T. (2008). All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC bioinformatics*, **9**(Suppl 11), S2.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *Computer Science*.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(8), 1798–1828.

Björne, J., Kaewphan, S., and Salakoski, T. (2013). Uturku: Drug named entity recognition and drug-drug interaction extraction using svm classification and domain knowledge. In *SemEval@ NAACL-HLT*, pages 651–659.

Bui, Q.-C., Sloot, P. M., Van Mulligen, E. M., and Kors, J. A. (2014). A novel feature-based approach to extract drug–drug interactions from biomedical text. *Bioinformatics*, **30**(23), 3365–3371.

Caruana, R. (1997). Multitask learning. *Machine Learning*, **28**(1), 41–75.

Chowdhury, M. F. M. and Lavelli, A. (2013a). Exploiting the scope of negations and heterogeneous features for relation extraction: A case study for drug-drug interaction extraction. In *2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.HLT-NAACL*, pages 765–771.

Chowdhury, M. F. M. and Lavelli, A. (2013b). Fbk-irst: A multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information. In *SemEval@ NAACL-HLT*, pages 351–355.

Faisal, M., Chowdhury, M., and Lavelli, A. (2013). Fbk-irst: a multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 351¨C–355, Atlanta, Georgia, USA. Association for Computational Linguistics.

Giuliano, C., Lavelli, A., and Romano, L. (2006). Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Conference of the European Chapter of the Association for Computational Linguistics*, volume 18, pages 401–408. Citeseer.

He, L., Yang, Z., Zhao, Z., Lin, H., and Li, Y. (2013). Extracting drug-drug interaction from the biomedical literature using a stacked generalization-based approach. *PLoS ONE*, **8**(6), e65814.

Hochreiter, S. and Schmidhuber, J. (1997). Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479.

Jari Björne, S. K. and Salakoski, T. (2013). Uturku: Drug named entity detection and drug-drug interaction extraction using svm classification and domain knowledge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 651¨C–659, Atlanta, Georgia, USA. Association for Computational Linguistics.

Kim, S., Liu, H., Yeganova, L., and Wilbur, W. J. (2015). Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach. *Journal of biomedical informatics*, **55**, 23–30.

Liu, S., Chen, K., Chen, Q., and Tang, B. (2016a). Dependency-based convolutional neural network for drug-drug interaction extraction. In *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pages 1074–1080. IEEE.

Liu, S., Tang, B., Chen, Q., and Wang, X. (2016b). Drug-drug interaction extraction via convolutional neural networks. *Computational and mathematical methods in medicine*, **2016**.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Moschitti, A. (2004). A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 335. Association for Computational Linguistics.

PK, H., DC, W., K, Z., DP, C., JC, M., and LR, C. (1993). Terfenadine-ketoconazole interaction: Pharmacokinetic and electrocardiographic consequences. *JAMA*, **269**(12), 1513–1518.

Quan, C., Hua, L., Sun, X., and Bai, W. (2016). Multichannel convolutional neural network for biological relation extraction. *BioMed research international*, **2016**.

Sahu, S. K. and Anand, A. (2017). Drug-drug interaction extraction from biomedical text using long short term memory network. *arXiv preprint arXiv:1701.08303*.

Segura Bedmar, I., Martinez, P., and Sánchez Cisneros, D. (2011). The 1st ddiextraction-2011 challenge task: Extraction of drug-drug interactions from

biomedical texts.

Segura-Bedmar, I., Martínez, P., and de Pablo-Sánchez, C. (2011a). A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents. *BMC bioinformatics*, **12**(2), S1.

Segura-Bedmar, I., Martinez, P., and de Pablo-Sánchez, C. (2011b). Using a shallow linguistic kernel for drug–drug interaction extraction. *Journal of biomedical informatics*, **44**(5), 789–804.

Segura Bedmar, I., Martínez, P., and Herrero Zazo, M. (2013). Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). Association for Computational Linguistics.

Thomas, P. E., Neves, M. L., Rocktäschel, T., and Leser, U. (2013). Wbi-ddi: Drug-drug interaction extraction using majority voting. In *SemEval@ NAACL-HLT*, pages 628–635.

Tikk, D., Solt, I., Thomas, P., and Leser, U. (2013). A detailed error analysis of 13 kernel methods for protein–protein interaction extraction. *BMC bioinformatics*, **14**(1), 12.

Yi, Z., Li, S., Yu, J., and Wu, Q. (2017). Drug-drug interaction extraction via recurrent neural network with multiple attention layers. *arXiv preprint arXiv:1705.03261*.

Zhao, Z., Yang, Z., Luo, L., Lin, H., and Wang, J. (2016). Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. *Bioinformatics*, **32**(22), 3444–3453.

Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. (2016). Attention-based bidirectional long short-term memory networks for relation classification. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 207.