Extracting Drug-Drug Interactions with Attention CNNs

Masaki Asada, **Makoto Miwa** and **Yutaka Sasaki**
Computational Intelligence Laboratory
Toyota Technological Institute
2-12-1 Hisakata, Tempaku-ku, Nagoya, Aichi, 468-8511, Japan
{sd17402, makoto-miwa, yutaka.sasaki}@toyota-ti.ac.jp

Abstract

We propose a novel attention mechanism for a Convolutional Neural Network (CNN)-based Drug-Drug Interaction (DDI) extraction model. CNNs have been shown to have a great potential on DDI extraction tasks; however, attention mechanisms, which emphasize important words in the sentence of a target-entity pair, have not been investigated with the CNNs despite the fact that attention mechanisms are shown to be effective for a general domain relation classification task. We evaluated our model on the Task 9.2 of the DDIExtraction-2013 shared task. As a result, our attention mechanism improved the performance of our base CNN-based DDI model, and the model achieved an F-score of 69.12%, which is competitive with the state-of-the-art models.

1 Introduction

When drugs are concomitantly administered to patients, the effects of the drugs may be enhanced or weakened, which may cause side effects. These kinds of interactions are called Drug-Drug Interactions (DDIs). Several drug databases, such as DrugBank (Law et al., 2014), Therapeutic Target Database (Yang et al., 2016), and PharmGKB (Thorn et al., 2013), have been provided to summarize drug and DDI information for researchers and professionals; however, many newly discovered or rarely reported interactions are not covered in the databases, and they are still buried in biomedical texts. Therefore, studies on automatic DDI extraction that extract DDIs from texts are expected to support maintenance of databases with high coverage and quick update to help medical experts deepen their understanding of DDIs.

For the DDI extraction, deep neural network-based methods have recently drawn a considerable attention (Liu et al., 2016; Zhao et al., 2016; Sahu and Anand, 2017). Deep neural networks have been widely used in the NLP field. They show high performance on several NLP tasks without requiring manual feature engineering. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are often employed for the network architectures. Among these, CNNs have an advantage that they can be easily parallelized and the calculation is thus fast with recent Graphical Processing Units (GPUs).

Liu et al. (2016) showed that CNN-based model can achieve a high accuracy on the task of DDI extraction. Sahu and Anand (2017) proposed an RNN-based model with attention mechanism to tackle the DDI extraction task and show the state-of-the-art performance. The integration of an attention mechanism into a CNN-based relation extraction is proposed by Wang et al. (2016). This is applied to a general domain relation extraction task SemEval 2010 Task 8 (Hendrickx et al., 2009). Their model showed the state-of-the-art performance on the task. CNNs with attention mechanisms, however, are not evaluated on the task of DDI extraction.

In this study, we propose a novel attention mechanism that is integrated into a CNN-based DDI extraction model. The attention mechanism extends attention mechanism by Wang et al. (2016) in that it deals with anonymized entities separately from other words and incorporates a smoothing parameter. We implement a CNN-based relation extraction model and integrate the novel mechanism into the model. We evaluate our model on the Task 9.2 of the DDIExtraction-2013 shared task (Segura Bedmar et al., 2013).

The contribution of this paper is as follows. First, this paper proposes a novel attention mechanism that can boost the performance on CNN-based DDI extraction. Second, the DDI extraction model with the attention mechanism achieves

Proceedings of the BioNLP 2017 workshop, pages 9–18,
Vancouver, Canada, August 4, 2017. ©2017 Association for Computational Linguistics

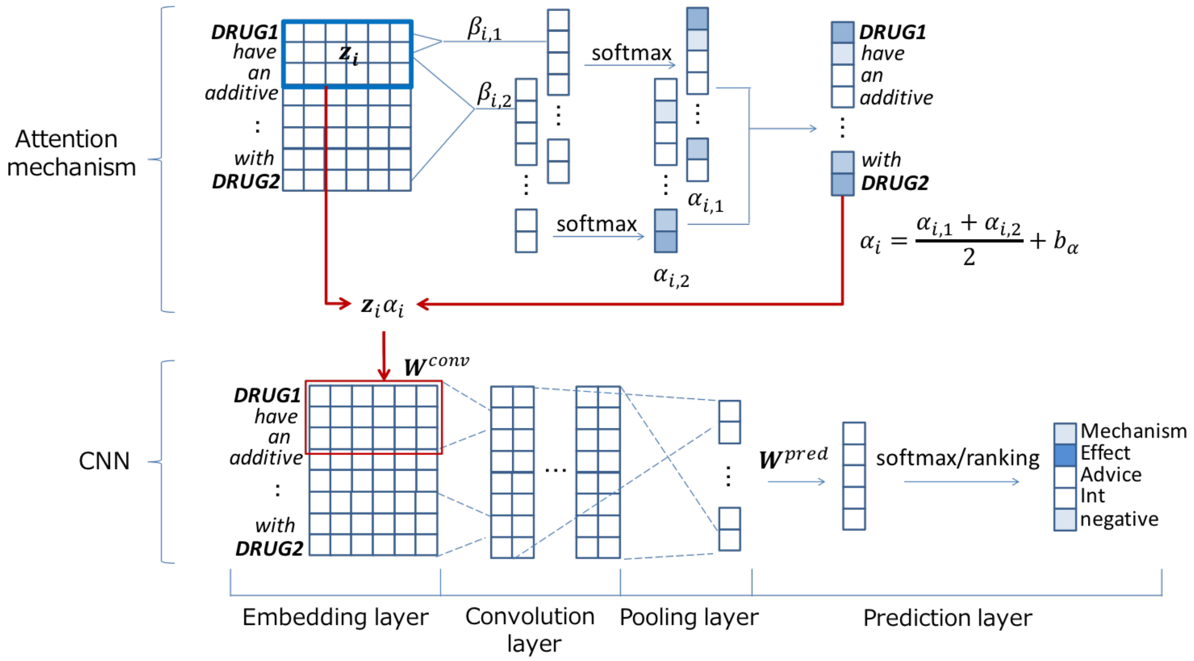

Figure 1: Overview of our model

the performance with an F-score of 69.12% that is competitive with other state-of-the-art DDI extraction models when we compare the performance without negative instance filtering (Chowdhury and Lavelli, 2013).

2 Methods

We propose a novel attention mechanism for a CNN-based DDI extraction model. We illustrate the overview of the proposed DDI extraction model in Figure 1. The model extracts interactions from sentences with drugs are given. In this section, we first present preprocessing of input sentences. We then introduce the base CNN model and explain the attention mechanism. Finally, we explain the training method.

2.1 Preprocessing

Before processing a drug pair in a sentence, we replace the mentions of the target drugs in the pair with "*DRUG1*" and "*DRUG2*" according to their order of appearance. We also replace other mentions of drugs with "*DRUGOTHER*".

Table 1 shows an example of preprocessing when an input sentence *Exposure to oral S-ketamine is unaffected by itraconazole but greatly increased by ticlopidine* is given with a target entity pair. By performing preprocessing, it is possible to prevent the DDI extraction model to be specialized for the surface forms of the drugs in a training data set and to perform DDI extraction using the information of the whole context.

2.2 Base CNN model

The base CNN model for extracting DDIs is one by Zeng et al. (2014). In addition to their original objective function, we employ an ranking-based objective function by dos Santos et al. (2015). The model consists of four layers: embedding, convolution, pooling, and prediction layers. We show the CNN model at the bottom half of Figure 1.

2.2.1 Embedding layer

In the embedding layer, each word in the input sentence is mapped to a real-valued vector representation using an embedding matrix that is initialized with pre-trained embeddings. Given an input sentence $S = (w_1, \cdots, w_n)$ with drug entities e_1 and e_2, we first convert each word w_i into a real-valued vector $\boldsymbol{w}_i^w$ by an embedding matrix $\boldsymbol{W}^{emb} \in \mathbb{R}^{d_w \times |V|}$ as follows:

$$\boldsymbol{w}_i^w = \boldsymbol{W}^{emb}\boldsymbol{v}_i^w, \tag{1}$$

where d_w is the number of dimensions of the word embeddings, V is the vocabulary in the training data set and the pre-trained word embeddings, and $\boldsymbol{v}_i^w$ is a one hot vector that represents the index of word embedding in $\boldsymbol{W}^{emb}$. $\boldsymbol{v}_i^w$ thus extracts the corresponding word embedding from $\boldsymbol{W}^{emb}$.

Entity1	Entity2	Preprocessed input sentence
S-ketamine	*itraconazole*	*Exposure to oral **DRUG1** is unaffected by **DRUG2** but greatly increased by DRUGOTHER.*
S-ketamine	*ticlopidine*	*Exposure to oral **DRUG1** is unaffected by DRUGOTHER but greatly increased by **DRUG2**.*
itraconazole	*ticlopidine*	*Exposure to oral DRUGOTHER is unaffected by **DRUG1** but greatly increased by **DRUG2**.*

Table 1: An example of preprocessing on the sentence "*Exposure to oral S-ketamine is unaffected by itraconazole but greatly increased by ticlopidine*" for each target pair.

The word embedding matrix $\boldsymbol{W}^{emb}$ is fine-tuned during training.

We also prepare d_{wp}-dimensional word position embeddings $\boldsymbol{w}^p_{i,1}$ and $\boldsymbol{w}^p_{i,2}$ that correspond to the relative positions from first and second target entities, respectively. We concatenate the word embedding $\boldsymbol{w}^w_i$ and these word position embeddings $\boldsymbol{w}^p_{i,1}$ and $\boldsymbol{w}^p_{i,2}$ as in the following Equation (2), and we use the resulting vector as the input to the subsequent convolution layer:

$$\boldsymbol{w}_i = [\boldsymbol{w}^w_i ; \boldsymbol{w}^p_{i,1} ; \boldsymbol{w}^p_{i,2}]. \tag{2}$$

2.2.2 Convolution layer

We define a weight tensor for convolution as $\boldsymbol{W}^{conv}_k \in \mathbb{R}^{d_c \times (d_w + 2d_{wp}) \times k}$ and we represent the j-th column of $\boldsymbol{W}^{conv}_k$ as $\boldsymbol{W}^{conv}_{k,j} \in \mathbb{R}^{(d_w+2d_{wp}) \times k}$. Here, d_c denotes the number of filters for each window size, k is a window size, and K is a set of the window sizes of the filters. We also introduce $\boldsymbol{z}_{i,k}$ that is concatenated k word embeddings:

$$\boldsymbol{z}_{i,k} = [\boldsymbol{w}^{\mathrm{T}}_{\lfloor i-(k-1)/2 \rfloor} ; \dots ; \boldsymbol{w}^{\mathrm{T}}_{\lfloor i-(k+1)/2 \rfloor}]^{\mathrm{T}}. \tag{3}$$

We apply the convolution to the embedding matrix as follows:

$$m_{i,j,k} = f(\boldsymbol{W}^{conv}_{k,j} \odot \boldsymbol{z}_{i,k} + b), \tag{4}$$

where $\odot$ is an element-wise product, b is the bias term, and f is the ReLU function defined as:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

2.2.3 Pooling layer

We employ the max pooling (Boureau et al., 2010) to convert the output of each filter in the convolution layer into a fixed-size vector as follows:

$$\boldsymbol{c}_k = [c_{1,k}, \cdots , c_{d_c,k}], \; c_{j,k} = \max_i m_{i,j,k}. \tag{6}$$

We then obtain the d_p-dimensional output of this pooling layer, where d_p equals to $d_c \times |K|$, by concatenating the obtained outputs $\boldsymbol{c}_k$ for all the window sizes $k_1, \cdots , k_K (\in K)$:

$$\boldsymbol{c} = [\boldsymbol{c}_{k_1} ; \dots ; \boldsymbol{c}_{k_i} ; \dots ; \boldsymbol{c}_{k_K}]. \tag{7}$$

2.2.4 Prediction layer

We predict the relation types using the output of the pooling layer. We first convert $\boldsymbol{c}$ into scores using a weight matrix $\boldsymbol{W}^{pred} \in \mathbb{R}^{o \times d_p}$:

$$\boldsymbol{s} = \boldsymbol{W}^{pred} \boldsymbol{c}, \tag{8}$$

where o is the total number of relationships to be classified and $\boldsymbol{s} = [s_1, \cdots , s_o]$. We then employ the following two different objective functions for prediction.

Softmax We convert $\boldsymbol{s}$ into the probability of possible relations $\boldsymbol{p}$ by a softmax function:

$$\boldsymbol{p} = [p_1, \cdots , p_o], \; p_j = \frac{\exp (s_j)}{\sum_{l=1}^{o} \exp (s_l)}. \tag{9}$$

The loss function $L_{softmax}$ is defined as in the Equation (10) when the gold type distribution $\boldsymbol{y}$ is given. $\boldsymbol{y}$ is a one-hot vector where the probability of the gold label is 1 and the others are 0.

$$L_{softmax} = -\sum \boldsymbol{y} \log \boldsymbol{p} \tag{10}$$

Ranking We employ the ranking-based objective function following dos Santos et al. (2015). Using the scores $\boldsymbol{s}$ in the Equation (8), the loss is calculated as follows:

$$\begin{aligned} L_{ranking} = & \log(1 + \exp(\gamma(m^+ - s_y)) \\ & + \log(1 + \exp(\gamma(m^- + s_c)), \end{aligned} \tag{11}$$

where m^+ and m^- are margins, γ is a scaling factor, y is a gold label, and c ($\neq y$) is a negative label with the highest score in $\boldsymbol{s}$. We set γ to 2, m^+ to 2.5 and m^- to 0.5 following dos Santos et al. (2015).

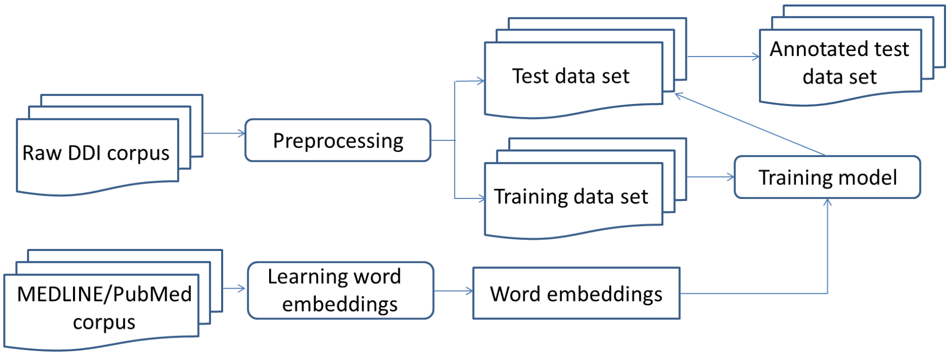

Figure 2: Workflow of DDI extraction

2.3 Attention mechanism

Our attention mechanism is based on the input attention by Wang et al. (2016)[1]. The proposed attention mechanism is different from the base one in that we prepare separate attentions for entities and we incorporate a bias term to adjust the smoothness of attentions. We illustrate the attention mechanism at the upper half of Figure 1.

We define the word index of the first and second target drug entities in the sentence as e_1 and e_2, respectively. We also denote by $E = \{e_1, e_2\}$ the set of indices and by $j \in \{1, 2\}$ the index of the entities. We calculate our attentions using these:

$$\beta_{i,j} = \boldsymbol{w}_{e_j} \cdot \boldsymbol{w}_i \tag{12}$$

$$\alpha_{i,j} = \begin{cases} \frac{\exp(\beta_{i,j})}{\sum_{1 \le l \le n, l \notin E} \exp(\beta_{l,j})}, & \text{if } i \notin E \\ a_{drug}, & \text{otherwise} \end{cases} \tag{13}$$

$$\alpha_i = \frac{\alpha_{i,1} + \alpha_{i,2}}{2} + b_\alpha. \tag{14}$$

Here, a_{drug} is an attention parameter for entities and b_α is the bias term. a_{drug} and b_α are tuned during training. If we set E to empty and b_α to zero, the attention will be the same as one by Wang et al. (2016). We incorporate the attentions α_i into the CNN model by replacing the Equation (4) with the following equation:

$$m_{i,j,k} = f(\boldsymbol{W}^{conv}_j \odot \boldsymbol{z}_{i,k} \alpha_i + b). \tag{15}$$

2.4 Training method

We use L2 regularization to avoid over-fitting. We use the following objective functions L'_* ($L'_{softmax}$ or $L'_{ranking}$) by incorporating the L2 regularization on weights to the Equation (10).

$$L'_* = L_* + \lambda(||\boldsymbol{W}^{emb}||^2_F + ||\boldsymbol{W}^{conv}||^2_F + ||\boldsymbol{W}^{pred}||^2_F) \tag{16}$$

Here, λ is a regularization parameter and $|| \cdot ||_F$ denotes the Frobenius norm. We update all the parameters including the weights $\boldsymbol{W}^{emb}$, $\boldsymbol{W}^{conv}$, and $\boldsymbol{W}^{pred}$, biases b and b_α, and the attention parameter a_{drug} to minimize L'_*. We use the adaptive moment estimation (Adam) (Kingma and Ba, 2015) for the optimizer. We randomly shuffle training data set and divide them into mini-batch samples in each epoch.

3 Experimental settings

We illustrate the workflow of the DDI extraction in Figure 2. As preprocessing, we performed word segmentation of the input sentences using the GENIA tagger (Tsuruoka et al., 2005). In this section, we explain the settings for the data sets, tasks, initial embeddings, and hyper-parameter tuning.

3.1 Data set

We used the data set from the DDIExtraction-2013 shared task (SemEval-2013 Task 9) (Segura Bedmar et al., 2013; Herrero-Zazo et al., 2013) for the evaluation. This data set is composed of documents annotated with drug mentions and their relationships. The data set consists of two parts: MEDLINE and DrugBank. MEDLINE consists of abstracts in PubMed articles, and DrugBank consists of the descriptions of drug interactions in the DrugBank database. This data set annotates the following four types of interactions.

- *Mechanism*: A sentence describes pharmacokinetic mechanisms of a DDI, e.g., "***Grepafloxacine*** *may inhibit the metabolism of* ***theobromine***."
- *Effect*: A sentence represents the effect of a DDI, e.g., "***Methionine*** *may protect against the ototoxic effects of* ***gentamicin***."

[1] We do not incorporate the attention-based pooling in Wang et al. (2016). We leave this for future work.

	Train		Test	
	DrugBank	MEDLINE	DrugBank	MEDLINE
No. of documents	572	142	158	33
No. of sentences	5,675	1,301	973	326
No. of pairs	26,005	1,787	5,265	451
No. of positive DDIs	3,789	232	884	95
No. of negative DDIs	22,216	1,555	4,381	356
No. of *Mechanism* pairs	1,257	62	278	24
No. of *Effect* pairs	1,535	152	298	62
No. of *Advice* pairs	818	8	214	7
No. of *Int* pairs	179	10	94	2

Table 2: Statistics for the DDIExtraction-2013 shared task data set

Parameter	Value
Word embedding size	200
Word position embeddings size	20
Convolutional window size	[3, 4, 5]
Convolutional filter size	100
Initial learning rate	0.001
Mini-batch size	100
L2 regularization parameter	0.0001

Table 3: Hyperparamters

	Counts
Sentences	1,404
Pairs	4,998
Mechanism pairs	232
Effect pairs	339
Advice pairs	132
Int pairs	48

Table 4: Statistics of the development data set

- *Advice*: A sentence represents a recommendation or advice on the concomitant use of two drugs, e.g., "***Alpha-blockers** should not be combined with **uroxatral**.*"

- *Int*: A sentence simply represents the occurrence of a DDI without any information about the DDI, e.g., "*The interaction of **omeprazole** and **ketoconazole** has established.*"

The statistics of the data set is shown in Table 2. As shown in this table, the number of pairs that have no interaction (negative pairs) is larger than that of pairs that have interactions (positive pairs).

3.2 Task settings

We followed the task setting of Task 9.2 in the DDIExtraction-2013 shared task (SemEval task 9). The task is to classify a given pair of drugs into the four interaction types or no interaction. We evaluated the performance with precision (P), recall (R), and F-score (F) on each interaction type as well as micro-averaged precision, recall, and F-score on all the interaction types. We used the official evaluation script provided by the task organizers and report the averages of 10 runs. Please note that we took averages of precision, recall and F-scores individually, so F-scores cannot be calculated from precision and recall.

3.3 Initializing embeddings

Skip-gram (Mikolov et al., 2013) was employed for the pre-training of word embeddings. We used 2014 MEDLINE/PubMed baseline distribution, and the size of vocabulary was 1,630,978. The embedding of the drugs, i.e., "*DRUG1*", "*DRUG2*" and "*DRUGOTHER*" are initialized with the pre-trained embedding of the word "*drug*". The embeddings of training words that did not appear in the pre-trained embeddings, as well as the word position embeddings, are initialized with the random values drawn from a uniform distribution and normalized to unit vectors. Words whose frequencies are one in the training data were replaced with an "*UNK*" word during training, and the embedding of words in the test data set that did not appear in both training and pre-trained embeddings were set to the embedding of the "*UNK*" word.

3.4 Hyperparameter tuning

We split the official training data set into two parts: training and development data sets. We tuned the hyper-parameters on the development data set on the softmax model without attentions. Table 3 shows the best hyperparameters on the softmax model without attentions. We applied the same

Type	P (%)	R (%)	F (%)
Softmax without attention			
Mechanism	76.24 (±4.48)	57.58 (±4.41)	65.31 (±1.76)
Effect	67.84 (±3.56)	63.61 (±4.95)	65.39 (±1.38)
Advice	82.26 (±7.04)	66.65 (±9.07)	72.75 (±2.72)
Int	**78.99** (±6.87)	33.55 (±2.62)	**47.05** (±1.71)
All (micro)	73.69 (±3.00)	59.92 (±3.73)	65.93 (±1.21)
Softmax with attention			
Mechanism	76.34 (±4.20)	**64.43** (±5.72)	67.86 (±4.10)
Effect	66.84 (±3.12)	65.98 (±2.63)	65.58 (±2.09)
Advice	80.98 (±6.14)	70.83 (±2.72)	76.28 (±1.40)
Int	73.21 (±6.30)	**38.44** (±9.82)	46.11 (±3.96)
All (micro)	73.74 (±1.88)	63.05 (±1.39)	67.94 (±0.70)
Ranking without attention			
Mechanism	78.41 (±3.99)	58.17 (±5.10)	66.51 (±2.61)
Effect	68.16 (±3.30)	65.75 (±3.22)	66.80 (±1.46)
Advice	**84.49** (±3.55)	67.14 (±4.68)	74.61 (±1.82)
Int	73.95 (±7.09)	33.43 (±1.18)	45.91 (±1.23)
All (micro)	74.79 (±2.41)	60.99 (±2.65)	67.10 (±1.09)
Ranking with attention			
Mechanism	**80.75** (±2.76)	61.09 (±3.03)	**69.45** (±1.45)
Effect	**69.73** (±2.64)	**66.63** (±2.93)	**68.05** (±1.29)
Advice	83.86 (±2.29)	**71.81** (±2.61)	**77.30** (±1.13)
Int	74.20 (±8.95)	33.02 (±1.40)	45.50 (±1.51)
All (micro)	**76.30** (±2.18)	**63.25** (±1.71)	**69.12** (±0.71)

Table 5: Performance of softmax/ranking CNN models with and without our attention mechanism. The highest scores are shown in bold.

hyperparameters to the other models. The statistics of our development data set is shown in Table 4. We set the sizes of the convolution windows to [3, 4, 5] that are the same as in Kim (2014). We chose the word position embedding size from $\{10, 20, 30, 40, 50\}$, the convolutional filter size from $\{10, 50, 100, 200\}$, the learning rate of Adam from $\{0.01, 0.001, 0.0001\}$, the mini-batch size from $\{10, 20, 50, 100, 200\}$, and the L2 regularization parameter λ from $\{0.01, 0.001, 0.0001, 0.00001\}$.

4 Results

In this section, we first summarize the performance of the proposed models and compare the performance with existing models. We then compare attention mechanisms and finally illustrate some results for the analysis of the attentions.

4.1 Performance analysis

The performance of the base CNN models with two objective functions, as well as with or without the proposed attention mechanism, is summarized in Table 5. The incorporation of the attention mechanism improved the F-scores by about 2 percent points (pp) on models with both objective functions. Both improvements were statistically significant ($p < 0.01$) with t-test. This shows that the attention mechanism is effective for both models. The improvement of F-scores from the least performing model (softmax objective function without our attention mechanism) to the best performing model (ranking objective function with our attention mechanism) is 3.19 pp (69.12% versus 65.93%), and this shows both objective function and attention mechanism are key to improve the performance. When looking into the individual types, ranking function with our attention mechanism archived the best F-scores on *Mechanism*, *Effect*, *Advice*, while the base CNN model achieved the best F-score on *Int*.

4.2 Comparison with existing models

We show comparison with the existing state-of-the-art models in Table 6. We mainly compare

Methods	*P* (%)	*R* (%)	*F* (%)
No negative instance filtering			
CNN (Liu et al., 2016)	75.29	60.37	67.01
MCCNN (Quan et al., 2016)	-	-	67.80
SCNN (Zhao et al., 2016)	68.5	61.0	64.5
Joint AB-LSTM (Sahu and Anand, 2017)	71.82	66.90	69.27
Proposed model	76.30	63.25	69.12
With negative instance filtering			
FBK-irst (Chowdhury and Lavelli, 2013)	64.6	65.6	65.1
Kim et al. (2015)	-	-	67.0
CNN (Liu et al., 2016)	75.72	64.66	69.75
MCCNN (Quan et al., 2016)	75.99	65.25	70.21
SCNN (Zhao et al., 2016)	72.5	65.1	68.6
Joint AB-LSTM (Sahu and Anand, 2017)	73.41	69.66	71.48

Table 6: Comparison with existing models

	P (%)	*R* (%)	*F* (%)
No attention	74.79 (±2.41)	60.99 (±2.65)	67.10 (±1.09)
Input attention by Wang et al. (2016)	73.48 (±1.96)	59.58 (±1.51)	65.77 (±0.80)
Our attention	76.30 (±2.66)	63.25 (±2.59)	69.12 (±0.71)
Our attention without separate attentions a_{drug}	74.03 (±2.11)	63.30 (±2.41)	68.17 (±0.71)
Our attention without the bias term b_α	71.56 (±2.18)	64.19 (±2.21)	67.62 (±0.96)

Table 7: Comparison of attention mechanisms on CNN models with ranking objective function

the performance without negative instance filtering, which omits some apparent negative instance pairs with rules (Chowdhury and Lavelli, 2013), since we did not incorporate it. We also show the performance of the existing models with negative instance filtering for reference.

In the comparison without negative instance filtering, our model outperformed the existing CNN models (Liu et al., 2016; Quan et al., 2016; Zhao et al., 2016). The model was competitive with Joint AB-LSTM model (Sahu and Anand, 2017) that was composed of multiple RNN models.

When considering negative instance filtering, our model showed lower performance than the state-of-the-art. However we believe we can get similar performance with theirs if we incorporate negative instance filtering. Still, the model outperformed several models such as Kim et al. (2015), Chowdhury and Lavelli (2013) and SCNN model even if we consider negative instance filtering.

4.3 Comparison of attention mechanisms

We compare the proposed attention mechanism with the input attention of Wang et al. (2016) to show the effectiveness of our attention mechanism. Table 7 shows the comparison of the attention mechanisms. We also show the base CNN-based model with ranking loss for reference, and the results of ablation tests. As is shown in the table, the attention mechanism by Wang et al. (2016) did not work in DDI extraction. However, our attention improved the performance. This result shows that the proposed extensions are crucial for modeling attentions in DDI extraction. The ablation test results show that both extensions to our attention mechanism, i.e., separate attentions for entities and incorporation of the bias term, are effective for the task.

4.4 Visual analysis

Figure 3 shows visualization of attentions on some sentences with DDI pairs using our attention mechanism. In the first sentence, "*DRUG1*" and "*DRUG2*" have a *Mechanism* interaction. The attention mechanism successfully highlights the keyword "*concentration*". In the second sentence, which have an *Effect* interaction, the attention mechanism put high weights on "*increase*" and "*effects*". The word "*necessary*" has a high weight on the third sentence with an *Advice* interaction. For an *Int* interaction in the last sentence, the word "*interaction*" is most highlighted.

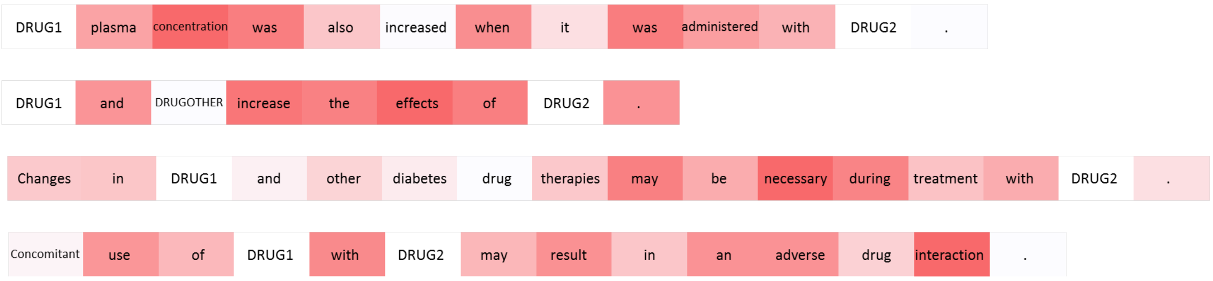

Figure 3: Visualization of attention

5 Related work

Various feature-based methods have been proposed during and after the DDIExtraction-2013 shared task (Segura Bedmar et al., 2013). Björne et al. (2013) tackled with DDI extraction using Turku Event Extraction System (TEES), which is an event extraction system based on the Support Vector Machines (SVMs). Thomas et al. (2013) and Chowdhury and Lavelli (2013) proposed two-phase processing models that first detected DDIs and then classified the extracted DDIs into one of the four proposed types. Thomas et al. (2013) used the ensembles of several kernel methods, while Chowdhury and Lavelli (2013) proposed hybrid kernel-based approach with negative instance filtering. The negative instance filtering is employed by all the subsequent models except for ours. Kim et al. (2015) proposed a two-phase SVM-based approach that employed a linear SVM with rich features including word features, word pairs, dependency relations, parse tree structures, and noun phrase-based constraint features. Our model does not use features and instead employs CNNs.

Deep learning-based models recently dominated the DDI extraction task. Among these, CNN-based models have been often employed and RNNs has received less attention. Liu et al. (2016) built a CNN-based model on word embedding and word position embeddings. Zhao et al. (2016) proposed Syntax CNN (SCNN) that employs syntax word embeddings with the syntactic information of a sentence as well as features of POS tags and dependency trees. Liu et al. (2016) tackled DDI extraction using Multi-Channel CNN (MCCNN) that enables the fusion of multiple word embeddings. Our work is different from theirs in that we employed an attention mechanism.

As for RNN-based approach, Sahu and Anand (2017) proposed an RNN-based model named Joint AB-LSTM (Long Short-Term Memory). Joint AB-LSTM is composed of the concatenation of two RNN-based models: bidirectional LSTM (Bi-LSTM) and attentive pooling Bi-LSTM. The model showed the state-of-the-art performance on the DDIExtraction-2013 shared task data set. Our model is a single model with a CNN and attention mechanism, and it performed comparable to theirs as shown in Table 6.

Wang et al. (2016) proposed muli-level attention CNNs and applied it to a general domain relation classification task SemEval 2010 Task 8 (Hendrickx et al., 2009). Their attention mechanism improved the macro F1 score by 1.9pp (from 86.1% to 88.0%), and their model achieved the state-of-the-art performance on the task.

6 Conclusions

In this paper, we proposed a novel attention mechanism for the extraction of DDIs. We built base CNN-based DDI extraction models with two different objective functions, softmax and ranking, and we incorporated the attention mechanism into the models. We evaluated the performance on the Task 9.2 of the DDIExtraction-2013 shared task, and we showed that both attention mechanism and ranking-based objective function are effective for the extraction of DDIs. Our final model achieved an F-score of 69.12% that is competitive with the state-of-the-art model when we compared the performance without negative instance filtering.

As future work, we would like to incorporate an attention mechanism in the pooling layer (Wang et al., 2016) and adopt negative instance filtering (Chowdhury and Lavelli, 2013) for the further performance improvement and fair comparison with the state-of-the-art methods.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 17K12741.

References

Jari Björne, Suwisa Kaewphan, and Tapio Salakoski. 2013. UTurku: drug named entity recognition and drug-drug interaction extraction using SVM classification and domain knowledge. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*. volume 2, pages 651–659.

Y-Lan Boureau, Jean Ponce, and Yann LeCun. 2010. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. pages 111–118.

Md Faisal Mahbub Chowdhury and Alberto Lavelli. 2013. FBK-irst: A multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information. *Atlanta, Georgia, USA* 351:53.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics*. volume 1, pages 626–634.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. SemEval-2010 Task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*. Association for Computational Linguistics, pages 94–99.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics* 46(5):914–920.

Sun Kim, Haibin Liu, Lana Yeganova, and W John Wilbur. 2015. Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach. *Journal of biomedical informatics* 55:23–30.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1746–1751.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR), San Diego, 2015.*

Vivian Law, Craig Knox, Yannick Djoumbou, Tim Jewison, An Chi Guo, Yifeng Liu, Adam Maciejewski, David Arndt, Michael Wilson, Vanessa Neveu, et al. 2014. DrugBank 4.0: shedding new light on drug metabolism. *Nucleic acids research* 42(D1):D1091–D1097.

Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. 2016. Drug-drug interaction extraction via convolutional neural networks. *Computational and mathematical methods in medicine* 2016.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Chanqin Quan, Lei Hua, Xiao Sun, and Wenjun Bai. 2016. Multichannel convolutional neural network for biological relation extraction. *BioMed Research International* 2016.

Sunil Kumar Sahu and Ashish Anand. 2017. Drug-drug interaction extraction from biomedical text using long short term memory network. *arXiv preprint arXiv:1701.08303* .

Isabel Segura Bedmar, Paloma Martínez, and María Herrero Zazo. 2013. SemEval-2013 Task 9: Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Proceedings of the 7th International Workshop on Semantic Evaluations*. Association for Computational Linguistics, pages 341–350.

Philippe Thomas, Mariana Neves, Tim Rocktäschel, and Ulf Leser. 2013. WBI-DDI: drug-drug interaction extraction using majority voting. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*. volume 2, pages 628–635.

Caroline F Thorn, Teri E Klein, and Russ B Altman. 2013. PharmGKB: the pharmacogenomics knowledge base. *Pharmacogenomics: Methods and Protocols* pages 311–320.

Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Junichi Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic Conference on Informatics*. Springer, pages 382–392.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics*. pages 1298–1307.

Hong Yang, Chu Qin, Ying Hong Li, Lin Tao, Jin Zhou, Chun Yan Yu, Feng Xu, Zhe Chen, Feng Zhu, and Yu Zong Chen. 2016. Therapeutic target database update 2016: enriched resource for bench to clinical drug target and targeted pathway information. *Nucleic acids research* 44(D1):D1069–D1074.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. 2014. Relation classification via convolutional deep neural network. In *COLING*. pages 2335–2344.

Zhehuan Zhao, Zhihao Yang, Ling Luo, Hongfei Lin, and Jian Wang. 2016. Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. *Bioinformatics* 32(22):3444–3453.