Bioinformatics, 37(12), 2021, 1739–1746
doi: 10.1093/bioinformatics/btaa907
Advance Access Publication Date: 24 October 2020
Original Paper

Data and text mining

Using drug descriptions and molecular structures for drug–drug interaction extraction from literature

Masaki Asada*, Makoto Miwa and Yutaka Sasaki

Toyota Technological Institute, 2-12-1 Hisakata, Tempaku-ku, Nagoya 468-8511, Japan

*To whom correspondence should be addressed.

Associate Editor: Elofsson Arne

Received on May 28, 2020; revised on October 7, 2020; editorial decision on October 8, 2020; accepted on October 9, 2020

Abstract

Motivation: Neural methods to extract drug–drug interactions (DDIs) from literature require a large number of annotations. In this study, we propose a novel method to effectively utilize external drug database information as well as information from large-scale plain text for DDI extraction. Specifically, we focus on drug description and molecular structure information as the drug database information.

Results: We evaluated our approach on the DDIExtraction 2013 shared task dataset. We obtained the following results. First, large-scale raw text information can greatly improve the performance of extracting DDIs when combined with the existing model and it shows the state-of-the-art performance. Second, each of drug description and molecular structure information is helpful to further improve the DDI performance for some specific DDI types. Finally, the simultaneous use of the drug description and molecular structure information can significantly improve the performance on all the DDI types. We showed that the plain text, the drug description information and molecular structure information are complementary and their effective combination is essential for the improvement.

Availability and implementation: Our code is available at https://github.com/tticoin/DESC_MOL-DDIE.

Contact: sd19501@toyota-ti.ac.jp

1 Introduction

When two or more drugs are administered to a patient at the same time, the effects of the drugs may be enhanced or weakened, which may also cause side effects. These kinds of interactions are called drug–drug interactions (DDIs). DDIs are reported in biomedical articles on a daily basis. Several drug databases, such as DrugBank (Wishart *et al.*, 2018), Therapeutic Target DB (Wang *et al.*, 2019) and PharmGKB (Whirl-Carrillo *et al.*, 2012), have been provided to integrate drug information including DDI information for researchers and professionals; however, not all interactions are registered in the databases, and valuable outcomes are still buried in biomedical articles. Therefore, automatic DDI extraction from biomedical literature is demanded.

Deep neural network-based DDI extraction methods have recently drawn a considerable attention because of their high performance. The methods require a large amount of text that is annotated by biomedical experts. Since the annotation efforts are costly and time-consuming, it is unrealistic to prepare a sufficient amount of annotated data. In addition, it is difficult to learn how to extract DDIs from text only with the limited amount of annotated text because deep understanding of DDI interaction descriptions often requires domain knowledge on drugs. Various drug information, such as detailed descriptions and molecular structure information of drugs, are registered in drug databases. Furthermore, models pre-trained on large-scale raw text show significant improvements in various natural language processing tasks (Devlin *et al.*, 2019). Effective use of such external information is necessary to reduce the reliance on annotated text.

In this study, we propose a method to utilize such external drug information in drug database DrugBank as well as large-scale raw text information for the extraction of DDIs from text. We focus on DrugBank because DDIExtraction 2013 shared task dataset is created based on the DrugBank database. We leave the incorporation of other databases for our future work. Specifically, we utilize the description and molecular structure information of drugs in the database. We also incorporate the information from large-scale raw texts by using a Bidirectional Encoder Representations from Transformers (BERT) model (Devlin *et al.*, 2019) pre-trained on large-scale raw text.

We illustrate the overview of the proposed method in Figure 1. For our baseline model, we employ the convolutional neural network (CNN)-based DDI extraction model (Asada *et al.*, 2018) that receives an input sentence with a target drug pair and classifies the pair into a specific DDI type. We enrich the input sentence using SciBERT (Beltagy *et al.*, 2019), which is a BERT model trained on large-scale biomedical and computer science text. We obtain the drug description representation of the target drugs using SciBERT and the molecular structure representation of the target drugs using molecular graph neural network (GNN) model proposed by

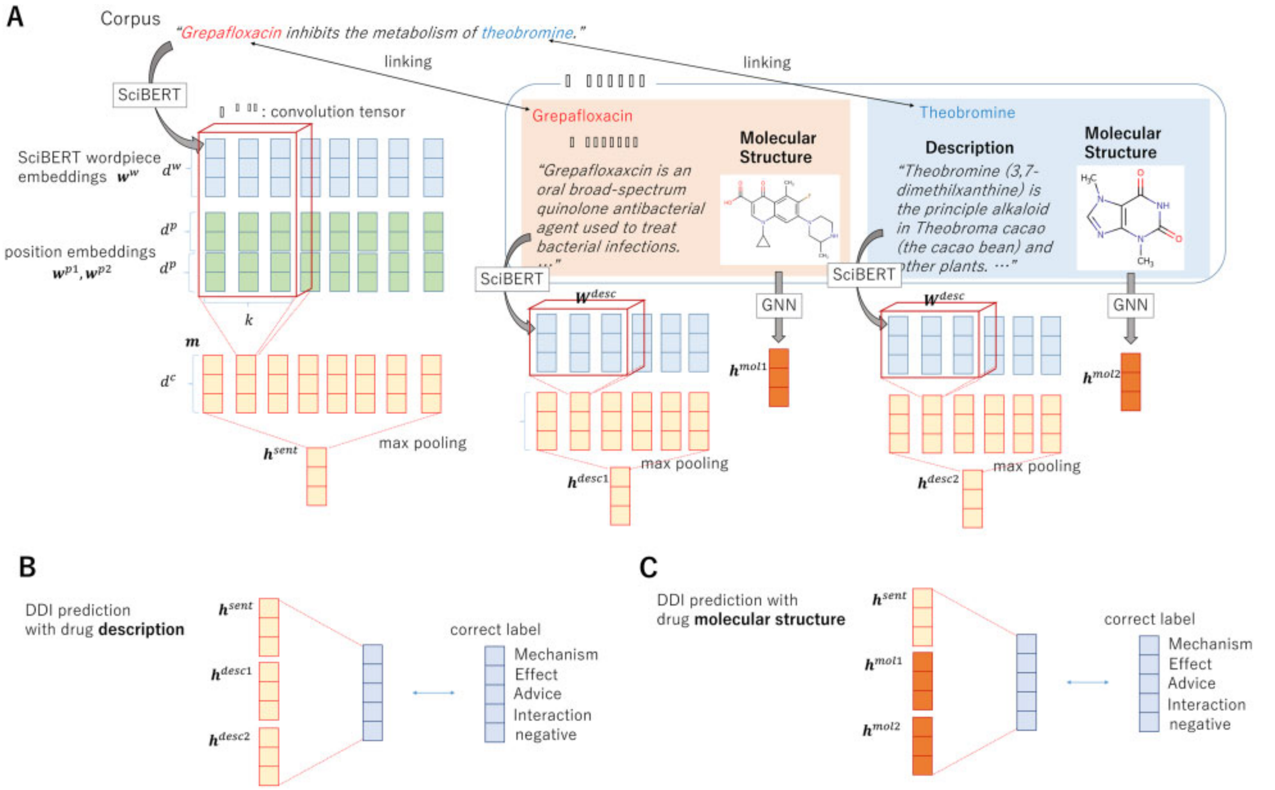

Fig. 1. Overview of our method. (**A**) Illustrates how to encode input sentences, drug descriptions and drug molecular structures. (**B**) and (**C**) show the prediction layer when the drug description representation and the drug molecular structure representation are used

Tsubaki *et al.* (2019). We combine these drug description and molecular structure representation with the enriched input sentence representation and classify the target drug pair into a specific DDI type.

We evaluated our method on the DDIExtraction 2013 shared task dataset (Segura-Bedmar *et al.*, 2013). Experimental results show that SciBERT boosts the performance of the baseline model. As a result, the performance is already strong enough and better than the previously reported performance. We show the drug database information is complementary to the large-scale pre-trained information, and the simultaneous use of drug description and drug molecular structure information can enhance the performance of DDI extraction from texts with SciBERT.

This article is a substantial extension of our work in ACL 2018 (Asada *et al.*, 2018) and we have following extensions:

- We replaced the token representation from word2vec to contextualized vectors obtained by SciBERT. As a result, we remarkably improved the performance of the baseline with the state-of-the-art performance.
- We employed the neural molecular GNN (Tsubaki *et al.*, 2019) that considers relatively large fragments of atoms and better represents molecular structures.
- We used drug descriptions registered in the drug database and we show drug description information is useful for extracting DDIs from corpus for some DDI types.
- We found the large-scale pre-training information, drug description and drug molecular information are complementary and their effective combination can largely improve the DDI extraction performance.

2 Related work

Various neural DDI extraction methods have been recently proposed using CNNs and recurrent neural networks since Liu *et al.* (2016) tackled the DDI extraction using the neural network-based method and outperformed various feature-based methods.

Especially in recent years, contextualized embeddings-based methods have been drawn a great attention (Peng *et al.*, 2019). Contextualized embeddings are pre-trained by a deep transformers-based method (Peng *et al.*, 2019) on large-scale text corpora. SciBERT is a model of the unsupervised pre-training method BERT (Devlin *et al.*, 2019), and it is pre-trained on a large multi-domain scientific corpus of Semantic Scholar (Ammar *et al.*, 2018). SciBERT achieved the state-of-the-art performance on several tasks in the biomedical domain, even compared with the bio-specific BioBERT (Peng *et al.*, 2019) model.

Several GNNs have been proposed for quantum chemistry, such as Duvenaud *et al.* (2015). In predicting drug properties, GNNs convert the molecular graph of a drug into a fixed-sized vector by aggregating the representation of atom nodes in the drug. Atoms in the drug are represented as nodes and bonds as edges. Tsubaki *et al.* (2019) proposed GNNs for molecular graphs, which takes subgraphs of the drug molecular graph as input, instead of single atoms. No GNN-based methods have been applied to the extraction of DDIs except for our previous work (Asada *et al.*, 2018).

3 DDI extraction

In this study, we propose a novel method to utilize drug database information for DDI extraction from text. We obtain the representation of input sentences by pre-trained contextualized embeddings, i.e. BERT, and CNNs. We link the mentions of target drugs to the drug entries in a drug database and acquire the description and

Table 1. An example of preprocessing

Mention1	Mention2	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****.*

Note: The input sentence contains three target drug pairs.

molecular structure information of these drugs. We also represent the drug descriptions by BERT and CNNs. We represent drug molecular structure by GNNs.

In this section, we first briefly overview the task setting of DDI extraction from texts. We then introduce the representations of input sentences, drug descriptions and drug molecular structures. We finally explain how to combine these representations to predict DDIs and train the model.

3.1 DDI extraction from text

DDI extraction is a task to identify drug pairs in an input sentence in which the interaction of the pairs is described and to assign the right types of interactions to the pairs. The task of extracting DDIs consists of two parts: named entity recognition and relation extraction (RE). In this study, we focus on the RE part, assuming drug entities are given, following existing methods (Liu *et al.*, 2016).

We treat the extraction of DDIs from text as a multi-class classification problem, where a part of target drug mentions and the remaining drug mentions are specified in the input sentence.

3.2 Input sentence representation

We follow the previous study (Liu *et al.*, 2016) to preprocess the input sentences. When three or more drug mentions appear in an input sentence, we duplicate the sentence for each drug mention pair. Specifically, if an input sentence contains n drug mentions, $\binom{n}{2}$ input sentences with different drug mention pairs are prepared. We preprocess each input sentence to specify the target drug mention pair and other drugs. In detail, we replace the target drug pair with DRUG1 and DRUG2 in the sentence order and replace other drugs with DRUGOTHER. We show the example of the preprocessing on the sentence *Exposure to oral S-ketamine is unaffected by itraconazole but greatly increased by ticlopidine* with different target drug pairs in Table 1.

We convert a preprocessed input sentence into a real-valued fixed-size vector by BERT and CNN-based model (Devlin *et al.*, 2019; Zeng *et al.*, 2014) and we show the model in the left part of Figure 1A. Given an input sentence $S = (w_1, \cdots, w_n)$ with drug mentions m_1 and m_2, we first split the sentence into wordpieces (a.k.a., subwords) by the WordPiece algorithm (Kudo and Richardson, 2018). We then convert each wordpiece w_i into a real-valued pretrained contextualized embedding $\boldsymbol{w}_i^w \in \mathbb{R}^{d^w}$ by the BERT model (light blue vectors in Fig. 1A). We also prepare d^p-dimensional position embeddings $\boldsymbol{w}_i^{p1}$ and $\boldsymbol{w}_i^{p2}$ for each wordpiece, which correspond to the relative positions from the first and second target mentions, respectively (green vectors in Fig. 1A). We concatenate the wordpiece embedding $\boldsymbol{w}_i^w$ and the position embeddings $\boldsymbol{w}_i^{p1}$ and $\boldsymbol{w}_i^{p2}$ as in the following Equation (1):

$$\boldsymbol{w}_i = [\boldsymbol{w}_i^w; \boldsymbol{w}_i^{p1}; \boldsymbol{w}_i^{p2}], \tag{1}$$

where [;] denotes concatenation. We use the resulting embeddings to prepare the input to the convolution layer.

We first introduce z_i that is the concatenation of k input embeddings (we can employ multiple windows instead of a single window with the size k, but we saw no significant difference in the performance in our preliminary experiment) around w_i:

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

We next apply convolution to the embeddings as follows:

$$m_{i,j} = f(W_j^{sent} \odot z_i + b^{sent}), \tag{3}$$

where $\odot$ is an element-wise product, b^{sent} is a bias term and $f(\cdot)$ is a GELU (Hendrycks and Gimpel, 2016) function (we chose the GELU activation function from ReLU, eLU, SeLU and GELU based on the results in our preliminary experiment). We define a weight tensor for convolution as $W^{sent} \in \mathbb{R}^{d^c \times (d^w + 2d^p) \times k}$. We represent the j-th column of W^{sent} as W_j^{sent}. k is a window size. We depict the tensor W^{sent} as a red box in the left part of Figure 1A. We then employ max-pooling to convert the output of each filter in the convolution layer into a fixed-size vector as follows:

$$\boldsymbol{h}^{sent} = \max_i m_{i,j}. \tag{4}$$

3.3 Drug description representation

Similarly to the input sentences, the description sentences of a drug mention are converted to the real-valued fixed-size vector by BERT and CNN. We directly use the wordpiece embeddings by BERT without word position embeddings to prepare the input to the convolution layer. We define a convolution weight tensor $W^{desc} \in \mathbb{R}^{d^c \times (d^w) \times k}$ and bias b^{desc} for description. Convolution and max-pooling are employed in the same way as the processing of the input sentences and we obtain the description representations $\boldsymbol{h}^{desc1}$ and $\boldsymbol{h}^{desc2}$ of drug mentions m_1 and m_2, respectively.

3.4 Molecular structure representation

We represent the molecular graph structures of drugs using GNNs. GNNs convert a drug molecule graph G into a fixed-size vector h^g. We represent atoms as nodes and bonds as edges in the graph. We employ the neural molecular GNN method proposed by Tsubaki *et al.* (2019). The molecular GNN method uses relatively large fragments referred to as r-radius subgraphs or molecular fingerprints to represent atoms with their contexts in the graph. The molecular GNN adopts fingerprint vectors as atom vectors, initializes the vectors randomly and updates them considering the graph structure of a molecule. We define the vector of the i-th atom in a drug molecule as $\boldsymbol{h}_i$ and the set of its neighboring atoms as N_i. The vector $\boldsymbol{h}_i$ is updated in the ℓ-th step as follows:

$$\boldsymbol{h}_i^{\ell} = \boldsymbol{h}_i^{\ell-1} + \sum_{j \in N_i} f(\mathbf{W}_{hidden}^{\ell-1} \boldsymbol{h}_j^{\ell-1} + \boldsymbol{b}_{hidden}^{l-1}), \tag{5}$$

where $f(\cdot)$ denotes a ReLU function. The drug molecular vector is obtained by summing up all the atom vectors and then the resulting vectors are fed into a linear layer.

$$\boldsymbol{h}^{mol} = f(\mathbf{W}_{output} \sum_i^M \boldsymbol{h}_i^L + \boldsymbol{b}_{output}), \tag{6}$$

where M is the number of fingerprints. Figure 2 shows how the molecular GNN model extracts fingerprints including β-lactam ($\boldsymbol{h}_1$) from penicillin drug ($r = 2$) and update fingerprint vectors.

We obtain the molecular structure representations $\boldsymbol{h}^{mol1}$ and $\boldsymbol{h}^{mol2}$ of drug mentions m_1 and m_2, respectively.

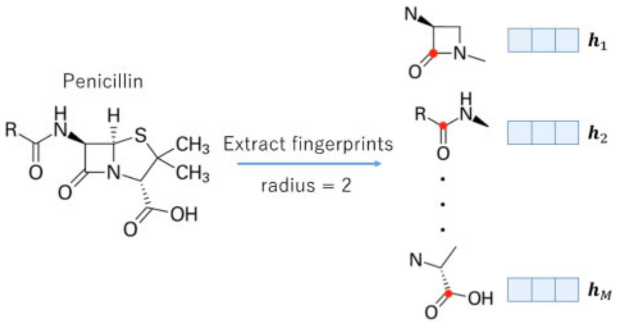

Fig. 2. Illustration of molecular fingerprints. This figure shows the extraction of several fingerprint subgraphs from a molecular structure when radius is 2

3.5 DDI extraction using database information

When we use the drug description information for DDI extraction, we concatenate the input sentence representation and two description representations as in Equation (7):

$$h = [h^{sent}; h^{desc1}; h^{desc2}]. \tag{7}$$

Similarly, two molecular structure representations are concatenated with the input sentence representation as in Equation (8):

$$h = [h^{sent}; h^{mol1}; h^{mol1}]. \tag{8}$$

We use the resulting vector as the input to the prediction layer. We convert h into prediction scores using a weight matrix $W^{pred} \in \mathbb{R}^{o \times d_p}$:

$$s = W^{pred} h, \tag{9}$$

where $s = [s_1, \ldots, s_o]$ and o is the number of DDI types. We convert s into the probability of possible interactions p by a softmax function:

$$p = [p_1, \ldots, p_o],\ p_j = \frac{\exp(s_j)}{\sum_{l=1}^{o} \exp(s_l)}. \tag{10}$$

We illustrates the DDI extraction using drug description information and drug molecular structure information in Figure 1B and C, respectively.

3.6 Training

The loss function L is defined as in Equation (11) using p in Equation (10) when the gold type distribution y is given. y is a one-hot vector where the probability of the gold label is 1 and the other probabilities are 0.

$$L = -\sum y \log p. \tag{11}$$

3.7 Ensemble

We employ an ensemble technique to combine the prediction from different models. Specifically, we simply sum up the prediction scores from different models for the ensemble after each of the models is trained separately. For instance, when we combine the prediction of the model with the description information and that with the molecular structure information, we sum up the prediction scores in Equation (9) as follows:

$$s = s^{desc} + s^{mol}. \tag{12}$$

Table 2. Statistics of SemEval-2013 dataset

	Train		Test	
	DrugBank	MEDLINE	DrugBank	MEDLINE
# documents	572	142	158	33
# sentences	5675	1301	973	326
# drug pairs	26 005	1787	5265	451
# positive pairs	3789	232	884	95
# negative pairs	22 216	1555	4381	356
Mechanism	1257	62	278	24
Effect	1535	152	298	62
Advice	818	8	214	7
Int.	179	10	94	2

4 Experimental settings

In this section, we explain the DDI extraction task settings, drug database preprocessing, drug mention linking and hyper-parameter settings.

4.1 DDI extraction task settings

We followed the DDIExtraction-2013 shared task (SemEval-2013 Task 9.2) (Segura-Bedmar *et al.*, 2013). This dataset is composed of documents annotated with drug mentions and their interactions. The dataset consists of two parts: MEDLINE and DrugBank. MEDLINE consists of abstracts in MEDLINE/PubMed articles, while DrugBank consists of the texts of drug interactions in the FDA label reference of DrugBank.

The task defines the following four interaction labels.

- *Mechanism*: this type is assigned when a pharmacokinetic mechanism is described in an input sentence.
- *Effect*: this type is assigned when the effect of the DDI is described.
- *Advice*: this is assigned when a recommendation or advice regarding the concomitant use of two drugs is described.
- *Int (Interaction)*: this type is assigned when the sentence simply states that an interaction occurs and does not provide any detailed information about the interaction.

A more detailed DDI type classification is directed to the annotation guidelines (https://www.cs.york.ac.uk/semeval-2013/task9/data/uploads/annotation_guidelines_ddi_corpus.pdf).

The statistics of the dataset with the official data split is shown in Table 2. Approximately 77% of the DDI corpus documents were randomly selected for the training dataset and rest were used for the test dataset by the official task organizers. This shows that the pairs with no interaction (negative pairs) are much more than the pairs with interactions (positive pairs).

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. While a macro-averaged metric is calculated by first calculating the metric for each type and then taking the average, a micro-averaged metric is calculated by directly calculating the metric for all the types.

4.2 DrugBank preprocessing

DrugBank is a freely available drug database containing more than 10 000 drugs. Each drug is given sentences describing its characteristics and efficacy. We show the first sentence of the drug description of *Salbutamo* as an example: *Salbutamol is a short-acting, selective beta2-adrenergic receptor agonist used in the treatment of asthma and COPD*. DrugBank also contains drug molecular structure information. Structure information is registered in SMILES string encoding.

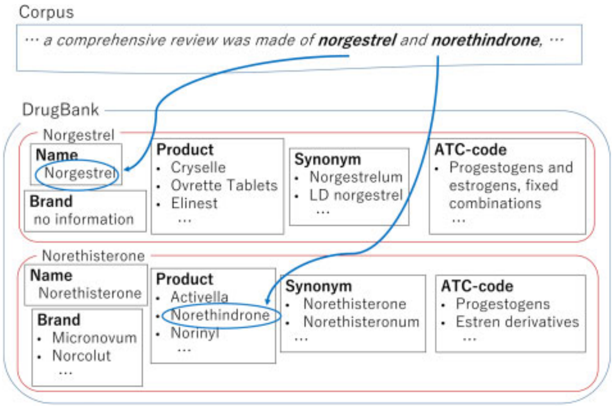

Fig. 3. Linking between mentions and DrugBank entry

To obtain the graph of a drug molecule, we took as input the SMILES string encoding of the molecule from DrugBank and then converted it into the graph structure using RDKit (Landrum, 2020). We extracted fingerprints from the graph using preprocessing scripts provided by Tsubaki *et al.* (2019).

4.3 Drug mention linking

We linked mentions in the corpus to DrugBank entries by relaxed string matching. In particular, we lowercased each mention and the following items in the DrugBank entries, and we chose the entry that includes an item showing the most overlap with the mention.

- Name: Headword of the drug entry
- Brand: Brand names from different manufactures
- Product: The final commercial preparation of the drug
- Synonym: Synonyms of the drug
- ATC codes: Codes for hierarchical drug classification.

For the ATC code, the same code can be assigned to multiple drugs, so we use only ATC codes that are assigned to single drugs for mention linking. Also, for synonyms, we linked mentions and synonyms by exact string matching instead of relaxed string matching to avoid the matching with very short strings (e.g. abbreviations). With this linking, 90.50% and 91.10% of drug mentions in SemEval-2013 train and test dataset matched the DrugBank entries. Figure 3 shows how the linking is performed. The input sentence contains two mentions 'norgestrel' and 'norethindrone'. We performed string matching to link these mentions to DrugBank entries. As a result, the mention 'noregestrel' matched the Name item and the mention 'norethindrone' matched the Product item.

4.4 Training settings

We followed the training settings for the fine-tuning of BERT on the GLUE tasks (Devlin *et al.*, 2019) except for the following two points. First, we employed the AdamW optimizer (Loshchilov and Hutter, 2019) instead of Adam optimizer. Second, we employed mixed-precision training (Le Gallo *et al.*, 2018) for the memory efficiency.

We applied dropout to the input of the convolution layer for regularization. Word position embeddings are initialized with random values drawn from a uniform distribution between -10^{-3} and 10^{-3}. We set the description and molecular structure vectors of unmatched entities to zero vectors. Tables 3 and 4 show hyper-parameters for CNNs and GNNs. We used the same hyper-parameters as the GLUE tasks in Devlin *et al.* (2019) for the BERT layer. In the DDIExtraction 2013 shared task, the official development dataset is not provided; thus we prepared a development dataset from the official training dataset to choose the other hyper-parameters. In order to train the model on the same setting as other existing models (Asada *et al.*, 2018; Liu *et al.*, 2016; Peng *et al.*, 2019), the development dataset is included in the entire training dataset for training the model. We used the entire training dataset for training the model to evaluate the performance on the test set. For GNNs, we show the results with different radii 0, 1 and 2 for molecular fingerprints. Note that, GNNs with a radius of 0 means no molecular fingerprints, which assigns vectors to atoms.

Table 3. Hyper-parameters for CNNs

Parameter	Value
Word embedding size d^w	768
Initial learning rate	5e-5
Number of fine-tuning epochs	3
L2 weight decay	0.01
Dropout rate	0.1
Mini-batch size	32
Word position embedding size d^p	10
Convolution window size k	5
Convolution filter size d^c	768
Convolution window size for description	3
Convolution filter size for description	20

Table 4. Hyper-parameters for GNNs

Parameter	Value
Molecular embedding size d^g	50
Number of hidden layer L	5
Radius	1

5 Results

Table 5 shows the performance of DDI extraction models including the proposed models with different settings and the state-of-the-art models. We can see that the baseline text-only model (SciBERT CNN) using SciBERT is powerful. SciBERT improved the performance of the model without SciBERT (word2vec CNN) by 11.04% points in the micro *F*-score. With this improvement, the model with SciBERT has achieved the state-of-the-art performance when we compare it with the state-of-the-art models in the top rows of the table. When we omitted the CNNs from the baseline model (SciBERT Linear), we used the first special token [CLS] as the aggregated representation of the sentence and we fed the embedding of [CLS] into the linear classifier layer. The performance slightly dropped with this omission but the difference is negligible. This indicates the BERT model is powerful enough to capture the similar information as CNNs.

We observe additional increase of the micro *F*-score by using drug description and molecular structure information as shown in the bottom part of the table. This shows the large-scale raw text information from SciBERT and the database information are complementary, and they are both useful for extracting DDIs from text. For GNNs, GNNs with molecular fingerprints (radius = 1 or 2) show better performance than GNNs without them (radius = 0), and GNNs with the radius of 1 show the highest performance. When comparing the description and molecular structure information, the micro *F*-score with molecular structure information (radius = 1) is slightly higher than one with the description information (+Desc), but their difference is not significant and the superiority depends on how to represent the molecular structure information, i.e. molecular fingerprints. We leave the search of the better representations for future work. The improvement by the ensemble model of description and the molecular structure information is statistically significant when compared with the baseline model ($P < 0.005$, McNemar test). We used the scikit-learn (Pedregosa *et al.*, 2011) Python library for evaluating the statistical significance.

Table 5. Evaluation on DDI extraction from texts on the test set

Method	*P*	*R*	*F* (%)
Liu *et al.* (2016)	75.29	60.37	67.01
BioBERT (Peng *et al.*, 2019)	—	—	78.8
Text-only (word2vec CNN) (Asada *et al.*, 2018)	71.97	68.44	70.16
Text-only (SciBERT linear)	80.28	81.92	81.09
Text-only (SciBERT CNN)	83.10	80.38	81.72
+ Desc	84.05	81.81	82.91
+ Mol (radius = 0)	83.29	82.02	82.65
+ Mol (radius = 1)	83.57	82.12	82.84
+ Mol (radius = 2)	83.66	81.10	82.36
+ Desc + Mol (radius = 1)	**85.36**	**82.83**	**84.08**
+ Desc + Mol (radius = 0,1,2)	84.51	82.53	83.51
+ Mol (radius = 0,1,2)	84.69	82.53	83.60

Note: We defined Text-only (SciBERT CNN) model as our baseline model. The best score is shown in bold.

Table 6. Evaluation on DDI extraction from texts on the development set

Method	*P*	*R*	*F* (%)
Text-only (SciBERT CNN)	83.55	80.19	81.84
+ Desc	83.19	82.31	82.75
+ Mol (radius = 0)	83.73	81.25	82.47
+ Mol (radius = 1)	82.85	83.90	83.37
+ Mol (radius = 2)	82.88	83.58	83.23
+ Desc + Mol (radius = 1)	**84.59**	**84.32**	**84.46**

Table 7. Performance on individual DDI types in *F*-scores

	DDI type			
Method	Mech.	Effect	Adv.	Int. (%)
Text-only	86.18	79.12	88.34	55.94
+ Desc	**87.62**	81.08	87.05	**60.27**
+ Mol (radius = 0)	84.65	81.20	90.67	55.71
+ Mol (radius = 1)	86.33	80.48	**92.07**	49.25
+ Mol (radius = 2)	84.02	**82.24**	88.58	57.34
+ Desc + Mol (radius = 1)	87.61	82.05	90.79	58.74

Note: The best score for each type is shown in bold and the scores lower than the baseline model are shown with underlines.

Table 6 shows the performance of DDI extraction models on the development dataset. Consistently with the results on the test set in Table 5, either of the description information and molecular structure information improves the performance and the combination of the two information showed the highest *F*-scores on the development dataset. However, there are some inconsistencies in the results on development and test datasets; the model with molecular structure information showed a higher *F*-score than the model with description information on the development dataset, while the model with molecular structure information showed a lower *F*-score on the test dataset.

Table 7 shows the *F*-scores on individual DDI types. The description information improves *F*-scores for *Mechanism*, *Effect* and *Int.* types, but it degrades the *F*-scores for *Advice*. The molecular structure information improves *F*-scores for *Effect* and *Advice*, but it degrades the *F*-scores for *Mechanism* and *Int.* for some radii. This indicates the two information have different effects on extracting DDIs, and each information is not enough to improve the entire DDI extraction performance. When both the description and molecular structure information are used by the ensemble technique, the model shows higher performance than the baseline model on all types. We cross-validated the training dataset using 5-fold cross-validation and we further analyzed the performance on individual DDI types. Table 8 shows the *F*-scores for folds of cross-validated training dataset. We used the micro-averaged *F*-score to calculate the average of the folds. The models with individual information show higher performance than the baseline model on *Mechanism* and *Int.*, while they show comparable or lower performance than the baseline model on other labels. Although the changes in performance are inconsistent for the DDI types and folds, the model with the ensemble technique shows higher performance than the models with individual information on average. As a result, the model with the ensemble technique improves the *F*-scores on average for all the types except for *Int.*, where our model performs on par with the baseline model. These results show that the

Table 8. Individual *F*-scores on 5-fold cross-validated training dataset

		DDI type			
	Method	Mech.	Effect	Adv.	Int. (%)
Fold 1	Text-only	84.60	**86.38**	85.80	68.29
	+ Desc	82.55	81.82	85.23	64.37
	+ Mol (radius = 1)	84.55	84.62	84.53	**71.05**
	+ Desc + Mol (radius = 1)	**86.13**	85.46	**86.69**	67.47
Fold 2	Text-only	83.46	83.26	78.80	**81.48**
	+ Desc	84.15	82.52	81.99	79.01
	+ Mol (radius = 1)	82.26	**83.45**	81.64	76.54
	+ Desc + Mol (radius = 1)	**84.29**	83.38	**82.64**	79.01
Fold 3	Text-only	84.91	59.21	**76.54**	91.43
	+ Desc	83.40	**88.31**	73.53	91.67
	+ Mol (radius = 1)	84.43	86.24	75.24	**94.44**
	+ Desc + Mol (radius = 1)	**86.09**	87.25	76.22	92.96
Fold 4	Text-only	76.81	81.56	78.01	79.45
	+ Desc	77.54	82.47	**79.65**	81.16
	+ Mol (radius = 1)	**78.17**	84.03	77.34	76.92
	+ Desc + Mol (radius = 1)	77.35	**85.15**	79.40	**83.33**
Fold 5	Text-only	81.97	81.76	**89.51**	76.54
	+ Desc	84.95	83.02	87.73	**81.48**
	+ Mol (radius = 1)	86.09	83.74	87.23	73.33
	+ Desc + Mol (radius = 1)	**86.26**	**84.91**	88.34	75.00
Average	Text-only	82.34	76.99	81.67	**79.07**
	+ Desc	83.09	84.39	81.27	78.09
	+ Mol (radius = 1)	82.47	83.57	81.60	78.97
	+ Desc + Mol (radius = 1)	**84.01**	**85.20**	**82.70**	78.99

Note: We used the micro-averaged *F*-score to calculate the average of the folds. The best score for each type is shown in bold and the scores lower than the baseline model are shown with underlines.

Table 9. Comparisons of *F*-scores on different parts of the test set

Method	MEDLINE	DrugBank	Overall (%)
Text-only (SciBERT CNN)	74.57	82.44	81.72
+ Desc	74.41	83.75	82.91
+ Mol (radius = 0)	75.00	83.41	82.65
+ Mol (radius = 1)	73.98	83.71	82.84
+ Mol (radius = 2)	74.57	83.15	82.36
+ Desc + Mol (radius = 1)	**78.16**	**84.67**	**84.08**

Table 10. Accuracy of binary classification on the DrugBank pairs

		Accuracy (%)
Description	SciBERT	91.05
Molecular structure	GNN (radius = 0)	67.58
	GNN (radius = 1)	82.21
	GNN (radius = 2)	89.36

performance on each label is affected by data splitting, but overall, when both the description information and molecular structure information are used by the ensemble technique, our model is effective for improving the performance of DDI extraction.

Table 9 shows the comparison of *F*-scores on the two different subsets of the test set: MEDLINE and DrugBank. The model with the description and one with molecular structure (radius = 1) degrade the *F*-score for MEDLINE, whereas both the description and molecular structure information improved the *F*-scores for DrugBank. For both subsets, the ensemble model greatly improved the *F*-score. These results also indicate the description and molecular structure information are complementary.

6 Discussion

6.1 Pre-training of GNNs and CNNs on DrugBank

To investigate the further use of DrugBank information, we verify if the DrugBank DDI labels can improve the DDI extraction performance. Specifically, we pre-trained GNNs for molecular structure information and CNNs for description information on DrugBank DDI labels. Many drug pairs have information of interactions, so this pre-training needs no additional annotations.

We extracted 50 000 interacting (positive) pairs from DrugBank. We note that, unlike the DDIExtraction 2013 shared task dataset, DrugBank only contains the information of interacting pairs; there are no detailed labels and no information for non-interacting (negative) pairs. We thus generated the same number of pseudo-negative pairs by randomly pairing drugs and removing those in positive pairs. To avoid overestimation of the performance, we deleted drug pairs mentioned in the test set of the text corpus in preparing the pairs. We split positive and negative pairs into 4:1 for train and test data, and we evaluated the classification accuracy using only the molecular information or only the description.

We first show the performance of the accuracy of binary classification on DrugBank DDI pairs in Table 10. The performance is surprisingly high, although the accuracy is evaluated on automatically generated negative instances. Overall, both drug description and molecular structure information can capture DDI information in DrugBank. In detail, the accuracy with drug description information is higher than that with molecular structure information. For molecular structure information, GNN with the radius of 2 shows the best performance. The difference in accuracy between radius 0 and 2 is 21.78% points, and this large difference shows the importance of capturing molecular fingerprints for DDI.

We pre-trained CNNs and GNNs using the DrugBank interaction labels including the pseudo-negative labels and fine-tuned them on the DDIExtraction 2013 dataset. Table 11 shows the comparison of the *F*-scores with or without pre-trainng. Unfortunately, for all the settings, the models with pre-training show lower performance than those without pre-training. This may be because the labels in the DDI extraction tasks are annotated depending on the context of the pairs and the labels can be inconsistent with labels in DrugBank and because the pseudo-negative examples are used in training instead of the real negative examples.

Table 11. Evaluation on DDI extraction from texts with or without pre-training of GNNs for the molecular structure and CNNs for the description

	Methods	*P*	*R*	*F* (%)
	SciBERT	83.10	80.38	81.72
w/ pre-training	+ Desc	**84.62**	79.26	81.85
	+ Mol (radius = 0)	82.69	81.00	81.83
	+ Mol (radius = 1)	84.51	80.28	82.34
	+ Mol (radius = 2)	82.36	80.28	81.74
w/o pre-training	+ Desc	84.05	81.81	**82.91**
	+ Mol (radius = 0)	83.29	82.02	82.65
	+ Mol (radius = 1)	83.57	**82.12**	82.84
	+ Mol (radius = 2)	83.66	81.10	82.36

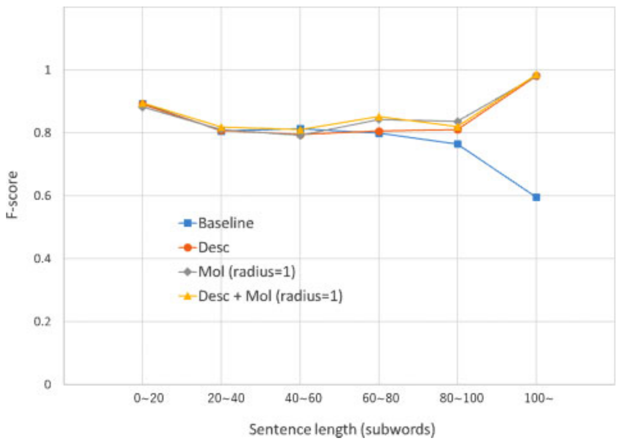

Fig. 4. *F*-scores for different sentence lengths on the 5-fold cross-validated training dataset. We used the micro-averaged *F*-score to calculate the average of the folds

6.2 Can DrugBank information alone extract DDIs from texts?

To further investigate how the contextual information is important in the DDI task, we verified whether the textual DDI can be extracted only from the drug information in DrugBank without using the input sentence. We simply omitted the input sentence representation h^{sent} from Equations (7) and (8) and trained the DDI extraction models, but the *F*-scores were quite low (<5%) for both models. This result shows that we cannot extract DDI relation from texts only with the description and molecular structure information. This indicates that DDI extraction from text greatly depends on the context information around drug mention pairs and our models on the database information serve as a supplement to the textual CNN model.

6.3 Error analysis

Figure 4 shows *F*-scores for different sentence lengths on the validation dataset. Since the instances with longer sentence lengths are relatively few, we used 5-fold cross-validation on the official training dataset. Here, the sentence length is defined to be the number of subwords divided by SciBERT vocabulary. In the previous work, Quan *et al.* (2016) analyzed the *F*-scores for the sentence length and pointed out that the performance is low for very long sentences with 60 or more words. Wang *et al.* (2017) also analyzed the *F*-scores for the sentence length and showed that *F*-scores tend to drop when the lengths of the instances are in the range from 71 to 100. The baseline model shows lower performance for long sentences with 80 or more subwords, and this result shows the same tendency as the previous analyses. Our model shows higher performance than the baseline model, especially for the sentences with more than 100 subwords. This shows that the DrugBank information is helpful to predict DDIs when the input sentences are long and complex and it is difficult to consider the whole contexts.

7 Conclusions

We proposed a novel neural method for DDI extraction from text using large-scale raw text information and drug database information, especially the drug descriptions and the drug molecular structure information. The results show that the large-scale raw text information with SciBERT greatly improves the performance of DDI extraction from text on the dataset of the DDIExtraction 2013 shared task. In addition, either of the drug description and the molecular structure information can further improve the performance for specific DDI types, and their simultaneous use can improve the performance on all the DDI types.

Our future work includes investigating other information registered in DrugBank and other drug databases. In addition, we will seek the way to build a model that can effectively utilize multiple items in drug databases and combine the textual and drug database information.

Funding

This work was supported by Japan Society for the Promotion of Science KAKENHI [17K12741, 20K11962].

Conflict of Interest: none declared.

References

Ammar,W. *et al.* (2018) Construction of the literature graph in semantic scholar. In: *Proceedings of NAACL-HLT 2018*. pp. 84–91. New Orleans, Louisiana.

Asada,M. *et al.* (2018) Enhancing drug-drug interaction extraction from texts by molecular structure information. In: *Proceedings of ACL 2018*. pp. 680–685. Melbourne, Australia.

Beltagy,I. *et al.* (2019) SciBERT: a pretrained language model for scientific text. In: *Proceedings of EMNLP-IJCNLP 2019*. pp. 3615–3620. Hong Kong, China.

Devlin,J. *et al.* (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of NAACL-HLT 2019*. pp. 4171–4186. Minneapolis, Minnesota.

Duvenaud,D.K. *et al.* (2015) Convolutional networks on graphs for learning molecular fingerprints. In: *Proceedings of NIPS 2015*. pp. 2224. Montréal, Canada.

Hendrycks,D. and Gimpel,K. (2016) Gaussian error linear units (GELUs). *arXiv Preprint arXiv: 1606.08415*.

Kudo,T. and Richardson,J. (2018) SentencePiece: a simple and language independent subword tokenizer and detokenizer for neural text processing. In: *Proceedings of EMNLP 2018*. pp. 66–71. Brussels, Belgium.

Landrum,G. (2020) RDKit: open-source cheminformatics software. http://www.rdkit.org, 2020.

Le Gallo,M. *et al.* (2018) Mixed-precision in-memory computing. *Nat. Electron.*, **1**, 246–253.

Liu,S. *et al.* (2016) Drug-drug interaction extraction via convolutional neural networks. *Comput. Math. Methods Med.*, **2016**, 1–8.

Loshchilov,I. and Hutter,F. (2019) Decoupled weight decay regularization. In: *Proceedings of ICLR 2019*. New Orleans, Louisiana, United States.

Pedregosa,F. *et al.* (2011) Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Peng,Y. *et al.* (2019) Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets. In: *Proceedings of BioNLP 2019*. pp. 58–65. Florence, Italy.

Quan,C. *et al.* (2016) Multichannel convolutional neural network for biological relation extraction. *BioMed Res. Int.*, **2016**, 1–10.

Segura-Bedmar,I. *et al.* (2013) SemEval-2013 task 9: extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In: *Proceedings of SemEval 2013*. pp. 341–350. Atlanta, Georgia, USA.

Tsubaki,M. *et al.* (2019) Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, **35**, 309–318.

Wang,W. *et al.* (2017) Dependency-based long short term memory network for drug-drug interaction extraction. *BMC Bioinformatics*, **18**, 578.

Wang,Y. *et al.* (2019) Therapeutic target database 2020: enriched resource for facilitating research and early development of targeted therapeutics. *Nucleic Acids Res.*, **48**, D1031–D1041.

Whirl-Carrillo,M. *et al.* (2012) Pharmacogenomics knowledge for personalized medicine. *Clin. Pharmacol. Ther.*, **92**, 414–417.

Wishart,D.S. *et al.* (2018) DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Res.*, **46**, D1074–D1082.

Zeng,D. *et al.* (2014) Relation classification via convolutional deep neural network. In: *Proceedings of COLING 2014*. pp. 2335–2344. Dublin, Ireland.