Journal of Biomedical Informatics 135 (2022) 104192

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Original Research

Extracting drug–drug interactions from no-blinding texts using key semantic sentences and GHM loss

Jiacheng Chen [a], Xia Sun [a,*], Xin Jin [a], Richard Sutcliffe [a,b,*]

[a] School of Information Science and Technology, Northwest University, Xi'an, 710127, China
[b] School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, UK

ARTICLE INFO

Keywords:
Drug–drug interactions
Drug blinding
Data imbalance
Label-noise

ABSTRACT

The extraction of drug–drug interactions (DDIs) is an important task in the field of biomedical research, which can reduce unexpected health risks during patient treatment. Previous work indicates that methods using external drug information have a much higher performance than those methods not using it. However, the use of external drug information is time-consuming and resource-costly. In this work, we propose a novel method for extracting DDIs which does not use external drug information, but still achieves comparable performance. First, we no longer convert the drug name to standard tokens such as DRUG0, the method commonly used in previous research. Instead, full drug names with drug entity marking are input to BioBERT, allowing us to enhance the selected drug entity pair. Second, we adopt the Key Semantic Sentence approach to emphasize the words closely related to the DDI relation of the selected drug pair. After the above steps, the misclassification of similar instances which are created from the same sentence but corresponding to different pairs of drug entities can be significantly reduced. Then, we employ the Gradient Harmonizing Mechanism (GHM) loss to reduce the weight of mislabeled instances and easy-to-classify instances, both of which can lead to poor performance in DDI extraction. Overall, we demonstrate in this work that it is better not to use drug blinding with BioBERT, and show that GHM performs better than Cross-Entropy loss if the proportion of label noise is less than 30%. The proposed model achieves state-of-the-art results with an F1-score of 84.13% on the DDIExtraction 2013 corpus (a standard English DDI corpus), which fills the performance gap (4%) between methods that rely on and do not rely on external drug information.

1. Introduction

The phenomenon of taking two or more drugs at the same time is common, because patients may suffer from more than one disease or need a drug combination to treat some conditions effectively [1]. However, some drug combinations may cause drug–drug interactions (DDIs) leading to serious health risks [2]. DDI extraction is an established field in which algorithms are developed to recognize DDIs in the published English medical literature.

In recent years, deep neural networks have been widely applied to extract DDIs. These methods can be divided into two types according to the embedding method used, namely the static word embedding-based method and the dynamic word embedding-based method. The static word embedding-based methods mainly use Word2Vec or GloVe word vectors to represent the sentence and utilize convolutional neural networks (CNNs) or recurrent neural networks (RNNs) to classify the DDI relations. Liu et al. [3] first proposed a CNN-based network to extract DDIs. Quan et al. [4] adopted a multichannel CNN and used richer vocabulary information to extract relations. In addition, Sahu et al. [5] proposed a long short-term memory (LSTM) network for DDI extraction and also achieved a good performance. Yi et al. [6] further utilized the Bidirectional Gated Recurrent Unit (BiGRU) and multiple attention layers to improve the ability to extract the DDIs. Zhang et al. [7] combined CNNs and RNNs to design a network to extract DDIs, and the results were higher than those based on only CNNs or RNNs. Sun et al. [8] not only used an LSTM and a hybrid CNN for DDI extraction but also utilized focal loss to alleviate the data imbalance problem in the DDI corpus. This achieved the best results based on static word vectors with an F1 value of 75.4%.

The dynamic word embedding-based methods mainly rely on pre-trained language models, such as BERT [9,10], BioBERT [11–13], and SCIBERT [14,15]. Li et al. [12] applied BioBERT and a graph convolutional network (GCN) to capture comprehensive contextual information and proposed a multi-task learning framework to alleviate the data imbalance problem. Peng et al. [10] used BioBERT alone,

[*] Corresponding authors.
E-mail addresses: raindy@nwu.edu.cn (X. Sun), rsutcl@nwu.edu.cn, rsutcl@essex.ac.uk (R. Sutcliffe).

https://doi.org/10.1016/j.jbi.2022.104192
Received 15 December 2021; Received in revised form 28 August 2022; Accepted 29 August 2022
Available online 3 September 2022

Table 1
All the similar instances generated from the original sentence ‘Barbiturates and glutethimide should not be administered to patients receiving coumarin drugs’.

Instance	Label	Prediction
Barbiturates and glutethimide should not be administered to patients receiving **coumarin drugs**.	advice	advice
Barbiturates and **glutethimide** should not be administered to patients receiving **coumarin drugs**.	advice	advice
Barbiturates and **glutethimide** should not be administered to patients receiving coumarin drugs.	**negative**	advice

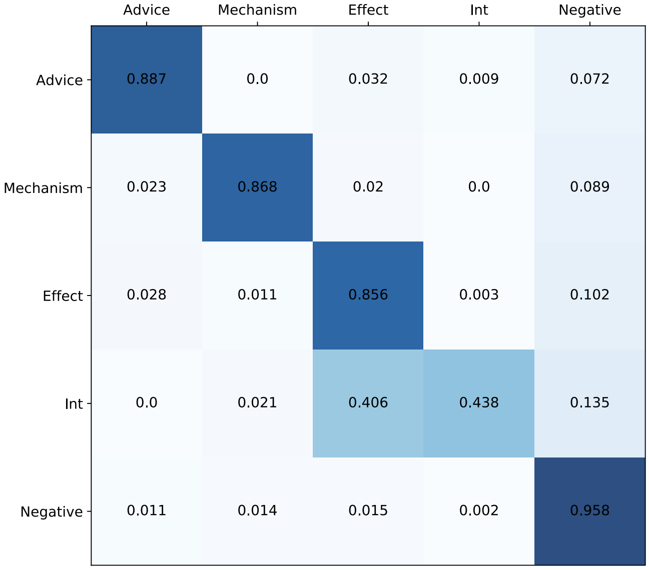

Fig. 1. The classification confusion matrix of BioBERT alone when applied to the DDIExtraction 2013 test corpus.

first pre-trained on a corpus collected from PubMed abstracts and clinical notes, to extract DDIs and achieved the best results without external drug information, reaching a 79.9% F1-score. Later, external drug information was utilized to further improve the performance of DDIs extraction methods. Zhu et al. [13] collected drug description information with a crawler from DrugBank and Wikipedia, and then used BioBERT to extract DDIs, reaching an 80.9% F1-score. Asada et al. [15] not only collected drug description information but also drug molecular structure information, and used SciBERT to achieve the current state-of-the-art result with an F1-score of 84.08%.

We can see that the dynamic word embedding-based methods show better performance than the static word embedding-based methods, and that external drug information has been utilized to further improve the performance. Although the use of external knowledge can improve the performance of extracting DDIs, it is time-consuming and resource-costly. Typically, such information needs to be collected with crawlers, converted into vectors, and combined with DDI instances.

In this work, we propose a dynamic word embedding-based method without external drug information which achieves state-of-the-art results with an F1-score of 84.13% on the DDIExtraction 2013 [16] corpus. This fills the performance gap (4%) between methods that rely on and do not rely on external drug information.

The DDIExtraction 2013 corpus is the standard dataset for the DDI extraction task. The corpus is in English and contains five types of DDI: four positive (*advice, effect, mechanism, int*) and one negative (*negative*). More detailed definitions and examples for each type are given in Section 3.1. Initially, BioBERT was applied to the corpus and a classification confusion matrix was created, as shown in Fig. 1. Each row represents the proportions of the corresponding DDI type which were classified into the types on the columns. From Fig. 1, we can firstly observe that the four kinds of *positive* instances (*advice, mechanism, effect, int*) are often misclassified into *negative* instances. It is noteworthy that the *negative* instances account for 80% of the corpus. Therefore, the *negative* instances are also easily misclassified as positive. This is similar to Zhu’s [13] observation. We found that these misclassified instances often have a very similar corresponding instance in the misclassified type. This is because the DDIExtraction 2013 corpus was constructed by collecting a list of complex sentences mentioning drugs, and then for each sentence in the list creating several training instances by marking up different pairs of drug entity instances and assigning a label accordingly. For example, Table 1 shows training instances generated from ‘Barbiturates and glutethimide should not be administered to patients receiving coumarin drugs.’ We can see that the labels of these similar instances are different: Two are *advice* type and the third is *negative* type, but they are all classified as *advice*. Here, the model tends to recognize the DDI relations of the selected drug entity pair based on the keywords in the instance, and ignores the selected drug entity pair. For example, the instances containing phrases such as ‘should not be administered’ are often classified as advice whatever the selected drug pair is. We call the misclassification of similar instances the *similar instance problem.*

We believe the similar instance problem can be alleviated by enhancing the selected drug entity pair and emphasizing the words closely related to the DDI relation of the selected drug pair. Therefore, we mark the selected drug pair by adding tags around the two selected drug entities ($ for the first and # for the second). Furthermore, we retain the drug entity name in the instance, which is always replaced by generic tokens in other works, a process called drug blinding. Finally, we input raw DDI instances (with $ and # around selected drug names) to BioBERT. We emphasize the words closely related to the DDI relation of the selected drug pair by using the Key Semantic Sentence (KSS) to retain only the keywords related to the selected drug entity pair by some grammatical dependency.

From Fig. 1, we can also observe that more than 40% of *int* instances are misclassified into *effect* type. An *int* instance only states that an interaction occurs and does not provide any additional information (e.g. ‘**FLEXERIL** may have life-threatening interactions with **MAO inhibitors**.’). Compared to the *int* instance, an *effect* instance will give more information about the effect of the interaction (e.g. ‘**TCAs** decrease the hypotensive effect of **guanfacine**.’). We found that the misclassification between effect instances and int instances is often caused by label noise, i.e. cases where instance labels are false. For example, the sentence **‘Barbiturates** may decrease the effectiveness of oral contraceptives, certain antibiotics, **quinidine**, theophylline, corticosteroids, anticoagulants, and beta blockers.’ describes the effect of the interaction between the selected drug pair. So the instance should be *effect*, but is labeled *int*. Previous researchers [13] also found that some of the instances of *int* type and *effect* type have similar semantics and about 10% of instances labeled as *int* are also labeled as *effect* in the DDIExtraction 2013 training set. We call the problem of poor DDI extraction performance due to incorrect labels the *label-noise problem.*

The label-noise problem is caused by errors made by biomedical domain experts when annotating the dataset. Some mistakes and inconsistencies are inevitable and these will reduce the performance of any DDI extraction model. We therefore adopt Gradient Harmonizing Mechanism (GHM) loss [17], a development of Focal loss [18], as our loss function to alleviate this problem. The GHM loss can decrease the weight of label-noise instances which fall in high-density gradient areas. At the same time, the DDI corpus also has the problem of data imbalance. Specifically, the number of instances of the five types (*negative*, *advice*, *effect*, *int*, and *mechanism*) in the corpus are 23,772,

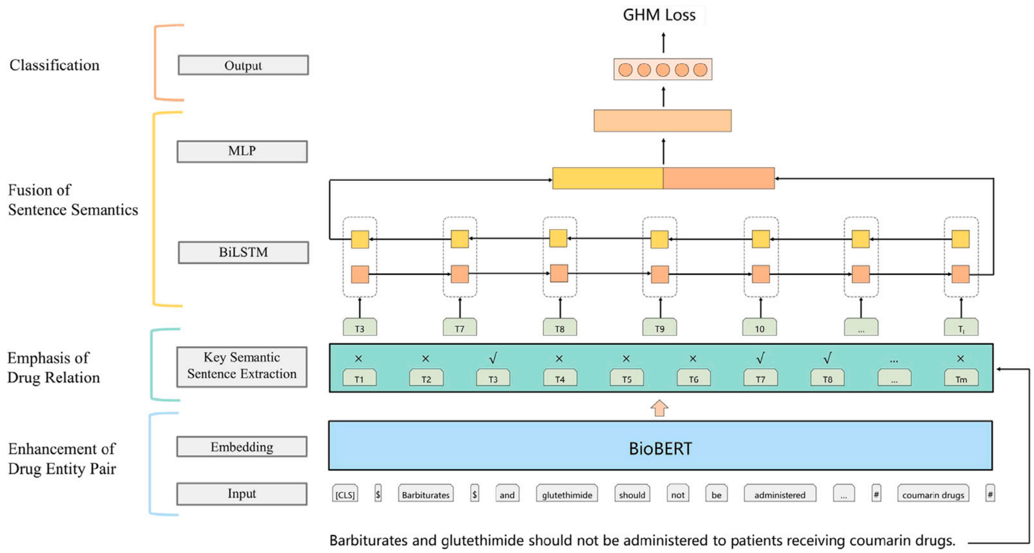

Fig. 2. The architecture of our model.

826, 1687, 188, and 1319 respectively. This imbalance leads to a bias towards classifying instances as *negative*. Fortunately, the GHM loss function can also reduce the weight of such instances, thereby alleviating the problem.

To summarize, this paper makes the following contributions:

- We explore the reasons why instances are misclassified when using BioBERT to extract DDIs, identify the *similar instance problem* and the *label-noise problem*, and propose corresponding solutions.
- We alleviate the similar instance problem by using full drug names with drug entity marking instead of blinding tokens, and by converting DDI sentences into KSSs.
- We address the label-noise and data imbalance problems together by using the GHM loss.
- We demonstrate that it is better not to use drug blinding with BioBERT, even though drug blinding is an effective technique when using earlier word embedding methods, such as Word2Vec and GloVe.
- We also verify the robustness of the GHM loss function to label noise and prove that GHM loss performs better than Cross-Entropy loss when the proportion of label-noise instances is lower than 30% in the training set.
- Overall, the proposed model achieves state-of-the-art results with an F1-score of 84.13% on the DDIExtraction 2013 corpus, which fills the performance gap (4%) between methods that rely on and do not rely on external drug information.

2. Methods

2.1. Model overview

We used a model with a conventional architecture (Fig. 2) to extract DDIs. There are four stages: (1) Preprocessing the text and encoding it using BioBERT, (2) Emphasizing the relations between the selected drug entity pair using KSS, (3) Fusing sentence semantics using a BiLSTM, and (4) Classifying the DDI type using an MLP.

We use the GHM loss function to train our model, to alleviate the label-noise and data imbalance problems by changing the weight of each instance.

2.2. Preprocessing

The aim is to extract all the DDIs in any sentence which has two or more drug entities. There are C_n^2 candidate DDI instances that will be generated from the same original sentence in total, where n is the number of drug entities. For example, for the sentence, ‘***Antacids** and **kaolin**: **Antacids** and **kaolin** can reduce the absorption of **chloroquine**.*’, there are five drug entities (in bold). A candidate DDI instance of *negative* type is ‘**Antacids** and **kaolin**: Antacids and kaolin can reduce absorption of chloroquine.’ and a *mechanism* type is ‘Antacids and kaolin: **Antacids** and kaolin can reduce absorption of **chloroquine**.’.

After preprocessing, we address the problem of data imbalance, using a *negative* instance filtering strategy. All instances which match one of the following manually-formulated rules are removed from the training data: (1) Two candidate drug entities have the same name or one drug is an abbreviation of the other; (2) Two candidate drug entities are in a coordinate structure; (3) Two candidate drug entities do not have a common father node in the dependency tree of the sentence. Rules (1) and (2) are proposed by other researchers [19,20], while Rule (3) is formulated by us.

2.3. Enhancement of drug entity pairs

As we stated in the introduction, we retain drug entity names in DDI sentences rather than using drug blinding. Then, we mark the two selected drug entities in each instance by using the ‘#’ and ‘$’ symbols to enhance the selected drug pair. For example, *$ **Barbiturates** $ and glutethimide should not be administered to patients receiving # **coumarin drugs** #*. Next, we use the WordPiece [9] tokenizer of BioBERT to split the words in the instance into subword segments (tokens), which is the standard procedure when using BioBert to encode a sentence. Using WordPiece gives a good balance between the flexibility of single characters and the efficiency of full words for decoding, and also sidesteps the need for special treatment of unknown words. Now, the instance becomes [$, Bar, ##bit, ##ura, ##tes, $, and, g, ##lut, ##eth, ##im, ##ide, should, not, be, administered, to, patients, receiving, #, co, ##uma, ##rin, drugs, #, .].

Formally, a sentence is represented by BioBERT as $S = [t_1, \ldots, t_\$, t_{d_1}, \ldots, t_{d_1}, t_\$, \ldots, t_\#, t_{d_2}, \ldots, t_{d_2}, t_\#, \ldots, t_m] \in R^{t_m * d_t}$, where t_i is the ith token in the sentence S, d_t is the length of the token vector representation, t_{d_1} or t_{d_2} is the token representation of the drug entity, and $t_\$$ or $t_\#$ is the token representation of the drug entity marking.

2.4. Key semantic sentences

After enhancing the selected pair of drug entities, we further emphasize the relations between those entities to alleviate the similar instance

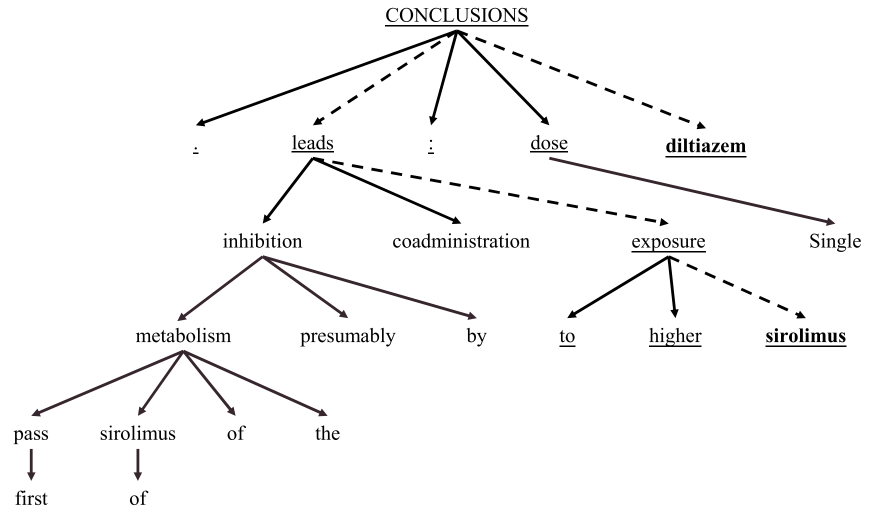

Fig. 3. The grammar dependency tree for the sentence 'CONCLUSIONS: Single-dose diltiazem coadministration leads to higher sirolimus exposure, presumably by inhibition of the first-pass metabolism of sirolimus.', shown with solid lines. The LCA subtree is marked in dashed lines, the words in the KSS are shown with underscore, and the candidate drug entities are in bold underscore. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

problem by using the KSS to delete inessential words. The construction of a KSS first requires a dependency tree to be built. The KSS is then extracted from the tree.

To construct the grammar dependency tree, the NLP dependency parser Stanza [21] is used to obtain the dependency word node h_i of each word w_i in a sentence. Then, we construct a grammatical dependency tree T in which each word w_i is treated as a node, and each dependency relation $w_i - h_i$ is treated as an edge. For example, the grammatical dependency tree for the sentence *'CONCLUSIONS: Single-dose diltiazem coadministration leads to higher sirolimus exposure, presumably by inhibition of the first-pass metabolism of sirolimus*.' is shown in Fig. 3.

After the dependency tree is constructed, we firstly search for the lowest common ancestor (LCA) of the two drug entity nodes. Then we determine the LCA subtree, which starts from a drug entity node, passes through its ancestor nodes, and ends at another drug entity node. Finally, the KSS is constructed by including the words that are up to one step away from the words in the LCA subtree. In Fig. 3 example, the LCA subtree of the selected drug entity is marked in solid red, and the words in the KSS are marked in dashed red. Finally, the sentence representation $S = [t_1, \dots, t_{\$}, t_{d_1}, \dots, t_{d_1}, t_{\$}, \dots, t_{\#}, t_{d_2}, \dots, t_{d_2}, t_{\#}, \dots, t_m] \in R^{t_m * d_t}$ is optimized by deleting all words which are not in the KSS. The result is $S_{KSS} = [t_1, \dots, t_l] \in R^{t_l * d_t}$, where l is the number of tokens after the sentence S is filtered by KSS.

We consider that words in the LCA subtree retain crucial information relating to the two-drug entities, while words one step away can add robustness to the model. In this way, KSS removes words which are inessential to the drug entities in order to emphasize the relations between the selected drug entity pair.

2.5. Fusion of sentence semantics

The S_{KSS} contains the retained token representations which not only enhance the selected drug entity pair but also match the relations between them. Then, the sentence semantics in the S_{KSS} are fused by the BiLSTM to obtain the forward sentence representation S_f and the reverse sentence representation S_b:

$$\begin{aligned} S_f &= \overrightarrow{LSTM}(S_{KSS}) \\ S_b &= \overleftarrow{LSTM}(S_{KSS}) \end{aligned} \tag{1}$$

where S_f, $S_b \in R^{d_{lstm}}$, and d_{lstm} is the hidden layer size of the LSTM. The sentence representations in the two directions are then fused through the MLP layer to obtain the final sentence representation S_F:

$$S_F = MLP([S_f; S_b]) \tag{2}$$

where $S_F \in R^{d_{mlp}}$, $R^{d_{mlp}}$ is the hidden layer size of the MLP, and [;] denotes the concatenation operation.

2.6. Classification

The final sentence representation S_F is fed into a fully-connected (FC) softmax layer to obtain the probability P of each DDI type:

$$p = softmax(WS_F + b) \tag{3}$$

where $W \in R^{d_r * d_{lstm}}$ and $b \in R^{dr}$ are the weight matrix and bias vector of the FC layer, and d_r is the number of the DDI type.

2.7. Model training

As we discussed in the introduction, there are two main problems in the DDI corpus, the label-noise problem and the data imbalance problem. To address both problems, we use GHM [17] as the loss function. This is an improved method based on Focal loss [18], which itself is derived from Cross-Entropy loss [22]. GHM readjusts the weight of each instance according to the gradient density, which is identified as the number of instances around the gradient. An easy-to-classify instance will produce a small gradient and a label-noise instance will produce a very large gradient in the model training. Li et al. [17] found that the instances with either very small gradient or very large gradient both have quite large gradient density. The GHM loss reduces the label-noise problem and data imbalance problem together by reducing the weight of such instances with high gradient density.

We formulate the GHM loss as:

$$GHM = \frac{1}{N} \sum_{i=1}^{N} L_{CE}(p_i, y_i)\beta_i \tag{4}$$

where N is the total number of training instances, i is the ith instance, L_{CE} is the CE loss which is calculated by the predicted probability p_i and the true probability y_i of each instance, and β_i is the weight of the ith instance. The β_i balances the importance of each instance, which is calculated by the gradient density $GD(g_i)$ of the ith instance:

$$\beta_i = \frac{N}{GD(g_i)} \tag{5}$$

Table 2
Statistics of the DDIExtraction 2013 dataset. 'Filtered' indicates that negative DDI instances are removed (see heuristics in 2.2). 'Proportion' is calculated by (Original-Filtered)/Original.

Types	Training			Test		
	Original	Filtered	Proportion	Original	Filtered	Proportion
DDI pairs	27792	19381	30.3%	5716	3831	33.0%
Positive	4020	3979	1.0%	979	972	0.7%
Negative	23772	15402	30.2%	4737	2859	39.6%
Advice	826	818	1.0%	221	221	0.0%
Effect	1687	1661	1.5%	360	357	0.8%
Int	188	187	0.5%	96	96	0.0%
Mechanism	1319	1313	0.5%	302	298	1.3%

where g_i is the gradient of the ith instance and $GD(g)$ is calculated as follows:

$$GD(g) = \frac{R_{ind(g)}}{\epsilon} \tag{6}$$

where $R_{ind(g)}$ is the number of examples lying in the region centered at g with a length of ϵ, and ϵ is the valid length of the region which could normalize the gradient density of g.

Intuitively, if i is a label-noise instance or *negative* instance (easy-to-classify), it will result in a very large or small gradient g_i, both of which correspond to a large gradient density $GD(g_i)$. Hence, a small β_i will be obtained. Therefore, the weight and importance of the instance i is reduced.

In summary, GHM loss can alleviate the label-noise problem and data imbalance problem together, without increasing the complexity of the model structure. Hence it can work together with BioBERT.

3. Experiments

3.1. Datasets

We evaluate our model on the DDIExtraction 2013 corpus [16] which is the standard dataset for the DDIE task. It contains 792 articles collected from DrugBank, and 233 abstracts collected from MEDLINE. There are five types of DDI: *advice, effect, mechanism, int, and negative*:

- *Advice* is used when a recommendation or advice is described (e.g. 'Concomitant use of **zalcitabine** and **lamivudine** is not recommended.').
- *Effect* is used when the effect of the DDI is described (e.g. '**TCAs** decrease the hypotensive effect of **guanfacine**.').
- *Mechanism* is used when a pharmacokinetic mechanism is described (e.g. '**Probenecid** competes with **meropenem** for active tubular secretion and thus inhibits the renal excretion of meropenem.').
- *Int* is used when the sentence simply states that an interaction occurs and does not provide any information about the interaction (e.g. '**FLEXERIL** may have life-threatening interactions with **MAO inhibitors**.').
- *Negative* is used when none of the above apply (e.g. 'The pharmacokinetics of **ethanol** were not affected by multiple-dose administration of **tiagabine**.').

The statistics of the original corpus and the filtered corpus are shown in Table 2. Two points should be noted. First, there is a serious data imbalance problem. Second, a large number of *negative* instances and a small number of *positive* instances are filtered by the *negative* instance filtering strategy. The filtered instances are equivalent to being predicated as *negative* instances.

3.2. Experimental settings

In our experiments, we use the PyTorch framework and the Transformers library [23] to implement our model, and the code is written in Python 3.6. We use a GeForce GTX TITAN X GPU to train and evaluate our model. We randomly select 10% of the training set as the development set to optimize the hyper-parameters which are listed in Table 3. It is worth mentioning that we use different learning rates for BERT and other parts to better coordinate and train the model. Bins and Alpha are hyper-parameters in the GHM loss.

Table 3
Hyper-parameters of our model.

Hyper-Parameter	Value
Batch_size	16
Max sequence length	384
Learning_rate_bert	2e−5
Learning_rate_other	1e−4
LSTM_hidden_size	768
MLP_hidden_size	300
Train_epochs	10
Adam_epsilon	1e−8
Dropout_rate	0.1
Bins	5
Alpha	0.75

We adopt micro-averaged F-score to evaluate our model, which is the official evaluation metric of the DDIExtraction 2013 task.

The micro-averaged F-score is defined as follows:

$$\begin{aligned} P_{\text{micro}} &= \frac{\sum_{n=1}^{N} TP_n}{\sum_{n=1}^{N} TP_n + \sum_{n=1}^{N} FP_n} \\ R_{\text{micro}} &= \frac{\sum_{n=1}^{N} TP_n}{\sum_{n=1}^{N} TP_n + \sum_{n=1}^{N} FN_n} \\ F_{\text{micro}} &= \frac{2 \times P_{\text{micro}} \times R_{\text{micro}}}{P_{\text{micro}} + R_{\text{micro}}}, \end{aligned} \tag{7}$$

where TP_n, FP_n, FN_n denote the true-positive, false-positive, and false-negative instance numbers of the nth class except *negative* type, respectively.

3.3. Experiment 1: Performance of proposed DDIE method

The aim was to compare the performance of the proposed model with eight existing models based on Word2Vec and four based on BERT. In particular, the two methods BioBERT(DD) and SciBERT(DD+DM) relied on external drug information. Results are shown in Table 4.

As can be seen, the proposed method achieved a 4% improvement in F1-score compared to the best drug-knowledge-free method (BioBERT, 79.90%), and even beyond the best method with drug knowledge (SciBERT(DD+DM), 84.08%) by a very narrow margin (0.05%). Relative to individual DDI types, the proposed model is 3.28% better than SciBERT(DD+DM) for *Effect* and 2.59% better for *Int*. This can be attributed to the fact that there are more label-noise instances in these DDI types than in the others. For *Advice*, the proposed method is 90.09%, 0.7% lower than SciBERT(DD+DM), and for *Mechanism* it is 84.21%, 3.4% lower. SciBERT(DD+DM) used molecular structure data for the drugs, information which is closely related to the drug *Mechanism*. This can account for the difference.

Table 4
Experiment 1: Performance comparison on the DDIExtraction 2013 corpus. The highest value in each column is shown in bold. 'DD' denotes the method utilizing drug description information. 'DM' denotes the method utilizing drug molecular structure information.

	Methods	F-score on each DDI type				Overall performance		
		Advice	Effect	Int	Mechanism	P	R	F
Word2Vec-based	CNN [3]	77.72	69.32	46.37	70.23	75.70	64.66	69.75
	MCCNN [4]	78.00	68.20	51.00	72.20	75.99	65.25	70.21
	Joint-LSTMs [5]	79.41	67.57	43.07	76.32	73.41	69.66	71.48
	GRU [6]	–	–	–	–	73.67	70.79	72.20
	CNN-GCNs [24]	81.62	71.03	45.83	73.83	73.31	71.81	72.55
	Recursive NN [25]	–	–	–	–	77.80	69.60	73.50
	RNN-CNN [7]	80.50	74.20	57.00	77.50	77.10	73.70	75.10
	RHCNN [8]	80.54	73.49	58.90	78.25	77.30	73.75	75.48
BERT-based	BioBERT-GCN [12]	–	–	–	–	77.60	75.70	76.60
	BioBERT [10]	–	–	–	–	–	–	79.90
	BioBERT(DD)[13]	86.00	80.10	56.60	84.60	81.00	80.90	80.90
	SciBERT(DD+DM)[15]	**90.79**	82.05	58.74	**87.61**	85.36	82.83	84.08
	Proposed method	90.09	**85.33**	**61.33**	84.21	**85.49**	**82.84**	**84.13**

Table 5
Results of Experiment 2: Ablation study. * marks significant differences between our method and ablations of the method with $p < 0.05$ under the McNemar test.

Ablation	P	R	F	△
Our method	85.49	82.84	84.13	
Our method with drug blinding	85.82	81.53	83.62	−0.51
Our method without KSS	83.71	82.86	83.28	−0.85
Our method without KSS and with drug blinding	81.91	83.16	82.53	−1.59*
Our method without drug entity marking	79.47	78.98	79.22	−4.91*
Our method without GHM	83.39	79.90	81.61	−2.52*

We also compare some specific differences between the SciBERT(DD+DM) method and our approach in terms of time and resources on the GeForce RTX 3090. For the inference time, our method (4ms/instance) is 2/3 of SciBERT(DD+DM) (6 ms/instance); for the number of parameters, our method (1.1 m) is 1/3 of SciBERT(DD+DM) (3.3 m), and for the consumption of memory, our method (12 GB) is 2/5 of SciBERT(DD+DM) (27 GB). The main difference between the approaches is that SciBERT(DD+DM) relies on external drug information. This requires an additional SciBERT to encode the drug description information, and a molecular GNN to encode the drug molecular structure information. As we can see, this results in a huge number of additional parameters, which take up more memory and result in longer inference times. In addition, the use of external drug information also requires considerable collection and preparation work in the early stages.

To investigate the proposed model further, the classification confusion matrix of our model (left) and the proportional change in the classification confusion matrix between our model and the one using BioBERT alone (right) are shown in Fig. 4.

We can see that our method improved the classification performance of almost all DDI types. The overall trend is that our method can correctly classify more *negative* instances into corresponding *positive* instances. Specifically, for *Advice*, *Mechanism*, *Effect* and *Int*, 63.6%, 50.0%, 67.5%, and 50.0% of the misclassified instances respectively are reduced. The reason is that we increase the discrimination between similar instances of *positive* and *negative* types.

In addition, the misclassification between *Advice* and *Effect* types has also been largely alleviated. The number of misclassification instances from *Advice* to *Effect* has dropped by 84.4%, while the misclassification from *Effect* to *Advice* has dropped by 89.3%.

3.4. Experiment 2: Ablation study

The aim was to find the contribution of No Blinding, KSS, and the GHM loss function to the performance of the proposed model by means of an ablation study. We also use the McNemar test to compare our complete method with ablations of the method in order to evaluate the statistical significance. Table 5 shows the results.

Table 6
Experiment 3: The performance changes of the model using word embedding methods with and without drug blinding technology. * marks significant differences with $p < 0.05$ under the McNemar test.

Embedding	Drug Blinding	F-score	Change
GloVe	✓	53.33	−8.87*
	✗	44.46	
BERT	✓	81.17	−1.18
	✗	80.99	
SciBERT	✓	83.28	−0.55
	✗	82.73	
BioBERT	✓	83.62	+0.51
	✗	84.13	

As can be seen, using either the original sentences or KSSs alone is beneficial, but the improvement is not significant, +0.51% and +0.85%, respectively. But when both the original sentences and KSSs are used, the improvement (+1.59%) is statistically significant when compared with the complete model ($p < 0.05$). This indicates that these two methods can complement each other in improving performance.

Moreover, the model using drug entity marking outperforms the model not using it by 4.91%, which is also statistically significant. This shows the necessity of using drug entity marking when not using drug blinding technology. If it is not used, the input instances derived from the same sentence will be the same, and the model will suffer from more serious similar instance problems, resulting in a drop in performance.

Finally, the model using GHM loss performs better than that using CE loss (+2.52%) which is once again statistically significant and shows that the mitigation of the label noise and data imbalance problems results in an improvement in the model.

3.5. Experiment 3: Word embeddings and drug blinding

The aim was to compare the performance of the proposed model when different word embeddings were incorporated, namely GloVe [26], BERT [9], SCIBERT [14] and BioBERT [11]. Moreover, these

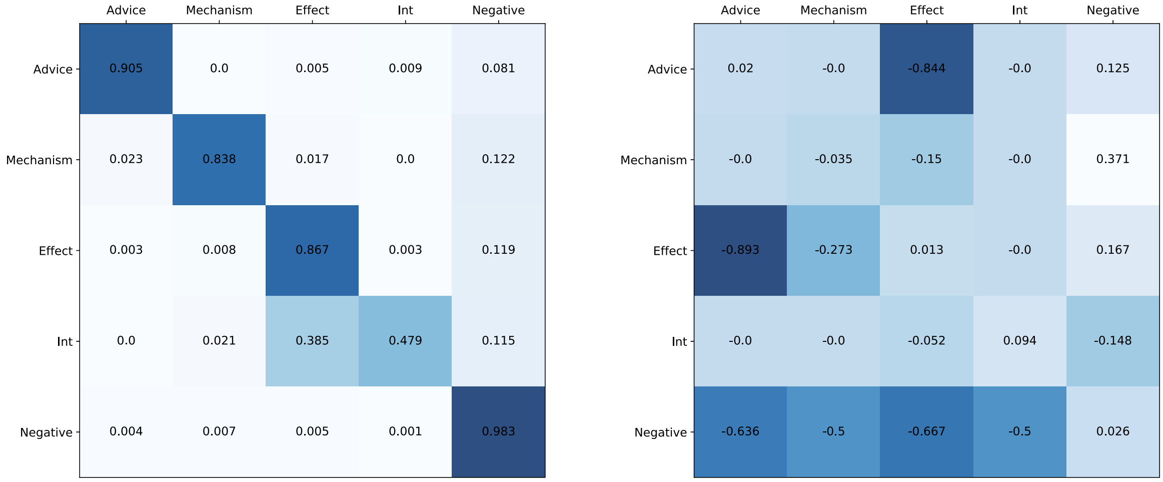

Fig. 4. Experiment 1: The classification confusion matrix of our method (left) and the proportional change in classification confusion matrix (right) of our method compared to BioBERT alone (Fig. 1), using the DDIExtraction 2013 test corpus.

Table 7
Experiment 4: The changes in F-score for CE and GHM loss with different noise ratios. * marks significant differences with $p < 0.05$ under the McNemar test.

Loss	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
CE	80.86	79.25	76.83	72.13	71.02	61.15	0	15.59	10.64	6.61
GHM	81.37	79.43	77.53	70.89	65.88	48.77	28.38	9.19	14.84	8.32
△	**0.51**	**0.18**	**0.7***	−1.24	−5.14*	−12.38*	**28.38***	−6.4*	**4.2***	**1.71***

embeddings were used both with and without drug blinding, and any change was tested for statistical significance using the McNemar test and $p < 0.05$. Table 6 shows the results.

According to the table, for all word embedding methods except BioBERT, those using drug blinding technology are better than those not using it. For GloVe, the performance of the model is greatly improved by using drug blinding (+8.87%) which is statistically significant. This is because models using traditional embeddings are more susceptible to complex and variable drug names.

However, for BioBERT, the performance when not using drug blinding technology is slightly better (+0.51%), presumably because BioBERT is pre-trained on a large-scale biomedical corpus. Therefore, the token representation of drug names contains drug knowledge similar to drug description information which is in the text, thus improving the performance of the model to identify DDI relationships.

3.6. *Experiment 4: Robustness of GHM loss for label-noise problem*

The aim was to investigate how the performance of the proposed model with GHM loss varies with the proportion of label-noise instances in the training set. GHM was also compared to CE loss and any changes were examined for significance. Once again, the McNemar test was used ($p < 0.05$).

We artificially changed the proportion of label-noise instances in the original data by modifying the label of each instance with a certain probability to create a series of different datasets. The model was then trained with each of these datasets in turn, and the performance evaluated. The same modified datasets were used to train and evaluate the model with a CE loss function. The results are shown in Table 7.

When the label-noise ratio is less than 0.3, the performance of the model is slightly better with GHM loss than it is with CE loss. Here, GHM loss is more robust to label noise than CE loss because it can reduce the weight of label-noise instances according to the gradient density of each instance.

When the label-noise ratio is between 0.4 and 0.6, the performance of the model is worse with GHM loss than it is with CE loss, and the difference is statistically significant ($p < 0.05$) when the ratio is 0.5 or 0.6. The reason may be that GHM relies on the model to give the correct gradient of the sample in order to work normally. When there are too many noise samples, the model's ability to generate the correct gradient of the sample decreases, resulting in poor performance.

When the label-noise ratio is greater than 0.6, the performance is poor and fluctuates constantly. Therefore, we could try to use other solutions that are specifically designed for solving the label-noise problem [27] to address this problem.

4. Conclusion

In this work, we identified two reasons why DDI instances are misclassified when using BioBERT to extract DDIs: the similar instance problem and the label-noise problem. To address these problems, we proposed a novel DDI extraction method, based on BioBERT, which consists of (1) No-Blinding that directly inputs the full drug names with drug entity marking to BioBERT so as to enhance the selected drug entity pair, (2) Key Semantic Sentences that enhance the relations between the selected drug entity pair to further alleviate the similar instance problem, and (3) GHM loss that changes the weight of each instance based on the gradient density to reduce the label-noise problem and data imbalance problem together. We conducted experiments on the DDIExtraction 2013 corpus and attained state-of-the-art results with an F-score of 84.13%, which fills the performance gap (4%) between methods that rely on external drug information and those that do not. In particular, the proposed model achieved the best results for *Effect* DDIs and *Int* DDIs: 85.33% and 61.33% F1-scores respectively. Finally, we verified the robustness of the GHM loss function.

In the future, we will explore how to combine our method with molecular structure information, which is important for the *Mechanism* type, and apply a more robust method to solve the label-noise problem. Finally, given the high cost of DDI dataset creation, we will study how to train a model on a small number of labeled instances to extract DDIs.

CRediT authorship contribution statement

Jiacheng Chen: Conceptualization, Methodology, Software, Writing – original draft. **Xia Sun:** Supervision, Funding acquisition. **Xin Jin:**

Visualization, Formal analysis. **Richard Sutcliffe:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This work has been supported by the Program for International Science and Technology Cooperation Projects of Shaanxi Province, China [2021KW-63].

References

[1] H. Liu, W. Zhang, B. Zou, J. Wang, Y. Deng, L. Deng, Drugcombdb: A comprehensive database of drug combinations toward the discovery of combinatorial therapy, Nucleic Acids Res. 48 (D1) (2020) D871–D881.

[2] L. Magro, U. Moretti, R. Leone, Epidemiology and characteristics of adverse drug reactions caused by drug–drug interactions, Expert Opin. Drug Saf. 11 (1) (2012) 83–94.

[3] S. Liu, B. Tang, Q. Chen, X. Wang, Drug-drug interaction extraction via convolutional neural networks, Comput. Math. Methods Med. 2016 (2016).

[4] C. Quan, L. Hua, X. Sun, W. Bai, Multichannel convolutional neural network for biological relation extraction, BioMed Res. Int. 2016 (2016).

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

[6] Z. Yi, S. Li, J. Yu, Y. Tan, Q. Wu, H. Yuan, T. Wang, Drug-drug interaction extraction via recurrent neural network with multiple attention layers, in: International Conference on Advanced Data Mining and Applications, Springer, 2017, pp. 554–566.

[7] Y. Zhang, H. Lin, Z. Yang, J. Wang, S. Zhang, Y. Sun, L. Yang, A hybrid model based on neural networks for biomedical relation extraction, J. Biomed. Inform. 81 (2018) 83–92.

[8] X. Sun, K. Dong, L. Ma, R. Sutcliffe, F. He, S. Chen, J. Feng, Drug-drug interaction extraction via recurrent hybrid convolutional neural networks with an improved focal loss, Entropy 21 (1) (2019) 37.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.

[10] Y. Peng, S. Yan, Z. Lu, Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMO on ten benchmarking datasets, 2019, arXiv preprint arXiv:1906.05474.

[11] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang, Biobert: A pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (4) (2020) 1234–1240.

[12] D. Li, H. Ji, Syntax-aware multi-task graph convolutional networks for biomedical relation extraction, in: Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis, LOUHI 2019, 2019, pp. 28–33.

[13] Y. Zhu, L. Li, H. Lu, A. Zhou, X. Qin, Extracting drug-drug interactions from texts with BioBERT and multiple entity-aware attentions, J. Biomed. Inform. 106 (2020) 103451.

[14] I. Beltagy, K. Lo, A. Cohan, Scibert: A pretrained language model for scientific text, 2019, arXiv preprint arXiv:1903.10676.

[15] M. Asada, M. Miwa, Y. Sasaki, Using drug descriptions and molecular structures for drug-drug interaction extraction from literature, Bioinformatics (2021).

[16] M. Herrero-Zazo, I. Segura-Bedmar, P. Martínez, T. Declerck, The DDI corpus: An annotated corpus with pharmacological substances and drug–drug interactions, J. Biomed. Inform. 46 (5) (2013) 914–920.

[17] B. Li, Y. Liu, X. Wang, Gradient harmonized single-stage detector, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, no. 01, 2019, pp. 8577–8584.

[18] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.

[19] S. Kim, H. Liu, L. Yeganova, W.J. Wilbur, Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach, J. Biomed. Inform. 55 (2015) 23–30.

[20] W. Zheng, H. Lin, L. Luo, Z. Zhao, Z. Li, Y. Zhang, Z. Yang, J. Wang, An attention-based effective neural model for drug-drug interactions extraction, BMC Bioinformatics 18 (1) (2017) 1–11.

[21] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C.D. Manning, Stanza: A python natural language processing toolkit for many human languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2020.

[22] R. Rubinstein, The cross-entropy method for combinatorial and continuous optimization, Methodol. Comput. Appl. Probab. 1 (2) (1999) 127–190.

[23] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer, et al., Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.

[24] M. Asada, M. Miwa, Y. Sasaki, Enhancing drug-drug interaction extraction from texts by molecular structure information, 2018, arXiv preprint arXiv:1805.05593.

[25] S. Lim, K. Lee, J. Kang, Drug drug interaction extraction from the literature using a recursive neural network, PLoS One 13 (1) (2018) e0190926.

[26] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1532–1543.

[27] H. Song, M. Kim, D. Park, Y. Shin, J.-G. Lee, Learning from noisy labels with deep neural networks: A survey, 2020, arXiv preprint arXiv:2007.08199.