TTI-COIN at BioCreative VII Track 1

Drug-protein interaction extraction with external database information

Naoki Iinuma, Masaki Asada, Makoto Miwa, and Yutaka Sasaki
Toyota Technological Institute, Nagoya, Aichi, Japan

***Abstract*— We propose two neural network-based methods that use external knowledge to predict drug-protein interactions in biomedical texts. In the first method, we construct denoised distant supervision data from external drug and protein databases. Then, we train a neural network model by adding the distant supervision data set to the annotated supervised data set. In the second method, we utilize the description and structure information registered in drug and protein databases.**

The experimental results show that the distant supervision data set improve F-scores for some interaction labels, although the overall micro-averaged F-score does not improve. The description and structure information is effective for the extraction of drug-protein interactions. Furthermore, the ensemble of the models mentioned above slightly enhanced the performance on the development data set.

Keywords—relation extraction, neural networks, database

I. Introduction

There is a growing need for systems that automatically detect in text chemical-protein interactions that are of crucial relevance for biology (1). The DrugProt task of the BioCreative VII Track 1 (2) aims to promote the development of systems that extract their interactions. We tackled the DrugProt task with neural models that employ external knowledge. Our models are based on BERT, which shows the state-of-the-art performance on several NLP tasks and can be considered external knowledge from other texts. In addition, we utilize distant supervision data and structural information of drugs and proteins as external knowledge from knowledge bases.

II. Task definition

The DrugProt task provides a corpus that domain experts annotated exhaustively, and all drug and protein mentions in the corpus are labeled. In addition, binary relationships corresponding to the 13 types of drug-protein interactions are annotated for all possible drug-protein pairs. In other words, when multiple binary relations exist, the drug-protein pair has multiple interactions. Conversely, when no binary relations exist, there is no interaction between the pair. The goal of the task is to correctly predict the interactions between drug-protein pairs given the input sentences and the mentions of drugs and proteins.

III. Methods

We propose two models and their ensembles; the first model utilizes the description and structural information of the protein and drug entities, while the second model considers distant supervision data. In this section, we first explain the preprocessing of the input data, which is common to all the models in section III *A*. We then explain the two models in section III *B* and section III *C* . We finally explain the ensemble method in section III *D*.

A. Preprocessing

The given abstract texts are first split into sentences by ScispaCy (3). Then, the drug-protein pairs are created from each sentence, and for each pair, the mentions of the target drug and protein are replaced with Entity1 and Entity2, respectively. Table I shows an example of this preprocessing. The sentence contains three drug mentions (*androstenedione*, *oestrone*, *oestrone*) and one protein mention (*aromatase*). Three drug-protein pairs are created from these with the mentions replaced.

B. Utilizing description and structural information of entities

The first model utilizes the description and structural information of the protein and drug entities. This model is based on the drug-drug interaction extraction method of Asada et al. (4), and we have extended the method for drug-protein interaction extraction. The drug and protein mentions in an input sentence are linked to the databases DrugBank (5) and Uniprot (6), respectively. The textual information and structural information registered in the databases are then used for relation extraction. Fig. 1 shows the overview of the model.

The mentions in the sentence and database entries are linked by relaxed string matching. For each drug, the "description" item registered in DrugBank is used as the description information, and the "SMILES" (7) item is used as structural information. For each protein, the "function" item registered in Uniprot is used as the description information, and the "sequence" item is used as the structural information.

1) Encoding input sentences

Each preprocessed input sentence is fed into a BERT encoder, and the embedding of [CLS] token is used as the input sentence representation vector $\boldsymbol{h}_{input}$. We then take $\boldsymbol{h}_{input}$ as the input of the fully connected layer and obtain the d_r -dimensional vector $\boldsymbol{h}_{input}^{fc}$, where d_r is the number of relation labels, including the negative label. It should be noted that although the DrugProt data set contains multiple labels for each instance, our approach cannot predict the multiple labels correctly. However, since it is known that the number of instances with multiple labels is relatively few in our preliminary experiment, we employ such a standard classification approach.

TABLE I. Examples of preprocessing of drug-protein pairs in the sentence "The aromatase enzyme, which converts androstenedione to oestrone, regulates the availability of oestrogen so support the growth of hormone-dependent beast tumours."

Entity1 (drug)	Entity2 (protein)	Preprocessed input sentence
androstenedione	*aromatase*	*The **Entity2** enzyme, which converts **Entity1** to oestrone, regulates the availability of oestrogen so support the growth of hormone-dependent beast tumours.*
oestrone	*aromatase*	*The **Entity2** enzyme, which converts androstenedione to **Entity1**, regulates the availability of oestrogen so support the growth of hormone-dependent beast tumours.*
oestrogen	*aromatase*	*The **Entity2** enzyme, which converts androstenedione to oestrone, regulates the availability of **Entity1** so support the growth of hormone-dependent beast tumours.*

We convert $\boldsymbol{h}_{input}^{fc}$ into the probability of possible relations by a softmax function $\boldsymbol{p}_{input} = \mathrm{softmax}(\boldsymbol{h}_{input}^{fc})$. The cross-entropy loss $L = \boldsymbol{y} \sum \log \boldsymbol{p}_{input}$ is used as the loss function, where $\boldsymbol{y}$ is the gold type distribution. $\boldsymbol{y}$ is a one-hot vector where the probability is 1 for the correct label and 0 otherwise.

2) Encoding description information

The descriptions registered in the database are also encoded by BERT in the same way as the input sentence of the corpus. A separate BERT is prepared for database descriptions. We convert the vector $\boldsymbol{h}_{desc\_CLS}$ of the BERT [CLS] token into a d_d-dimensional vector $\boldsymbol{h}_{desc}$ as follows:

$$\boldsymbol{h}_{desc} = \mathrm{GELU}\left(\boldsymbol{W}\boldsymbol{h}_{desc\_CLS} + \boldsymbol{b}\right), \tag{1}$$

where the GELU is an activation function, and $\boldsymbol{W}$ and $\boldsymbol{b}$ are weights and bias of the linear layer, respectively. We concatenate the representations of the entity 1 description $\boldsymbol{h}_{desc1}$ and the entity 2 description $\boldsymbol{h}_{desc2}$ and the input sentence $\boldsymbol{h}_{input}$. We then used the resulting vector as the input to the fully connected layer:

$$\boldsymbol{h}_{desc}^{fc} = \mathrm{FC}\left([\boldsymbol{h}_{input}; \boldsymbol{h}_{desc1}; \boldsymbol{h}_{desc2}]\right), \tag{2}$$

where FC is a fully connected layer and [;] denotes the vector concatenation. We convert $\boldsymbol{h}_{desc}^{fc}$ into the probability $\boldsymbol{p}_{desc}$ by a softmax function, and the model parameters are updated by minimizing the loss function $L = \boldsymbol{y} \sum \log \boldsymbol{p}_{desc}$.

3) Encoding structural information

For drugs, we use the SMILES strings as the structural information. For proteins, we use amino acid sequences. Both SMILES strings and amino acid sequences are encoded by character-based CNNs.

First, we assign the d_c-dimensional character embedding to each character of the structural sequence; specifically, atoms of drugs such as 'C' and 'N', or bonds of drugs such as '=' and '#', amino acid symbols of proteins such as 'A', 'R', and 'N'.

After each character of the sequence is converted to the corresponding embedding, all character embeddings are encoded as the inputs to CNNs with multiple convolutional window sizes (8), and max pooling is employed to obtain the whole sequence representation.

We concatenate the representation of the entity 1 structural representation $\boldsymbol{h}_{struct1}$, the entity 2 structural representation $\boldsymbol{h}_{struct2}$, and the input sentence representation $\boldsymbol{h}_{input}$ to make the input of the fully connected layer:

$$\boldsymbol{h}_{struct}^{fc} = \mathrm{FC}\left([\boldsymbol{h}_{input}; \boldsymbol{h}_{struct1}; \boldsymbol{h}_{struct2}]\right), \tag{3}$$

We convert $\boldsymbol{h}_{struct}^{fc}$ into the probability $\boldsymbol{p}_{struct}$ by a softmax function, and update the model parameters by minimizing the cross-entropy loss.

4) Inference

We finally combine description and structure information using an ensemble technique when predicting the drug-protein relation label.

The final prediction is obtained by averaging the prediction probabilities of the three models described in the previous section as follows:

$$\boldsymbol{p}_{all} = \frac{1}{3}\left(\boldsymbol{p}_{input} + \boldsymbol{p}_{desc} + \boldsymbol{p}_{struct}\right). \tag{4}$$

We calculate the relation label prediction as $\mathrm{argmax}(\boldsymbol{p}_{all})$.

C. Methods with distant supervision data

We tackled a method for training neural network models by adding distant supervision data constructed from databases to the DrugProt dataset.

1) Constructing distant supervision data

The flow of the creation of the distant supervision data is shown in Fig. 2. Four databases are used to create the distant supervision data: the drug database DrugBank, the protein database UniProt, the chemical substance database CTD (9), and the medical literature database PubMed (10).

First, relation triples are obtained from DrugBank. Relation triples are triples of IDs of interacting drugs and proteins and their interaction names. Next, a drug name dictionary is assembled by mapping drug IDs to surface expressions based on the information in DrugBank and CTD. Similarly, a protein name dictionary is assembled from UniProt and CTD. Then, we extract texts from the PubMed literature, split the extracted texts into sentences, and extract entities from the sentences using ScispaCy. Finally, we create distant supervision data by dictionary matching of drugs and proteins in the relation triples to the extracted entities from the PubMed using the drug name dictionary and protein name dictionary. The mapping between the relations on DrugBank and those on the task is done by using a dictionary created based on the descriptions of the relations in the DrugProt corpus relation annotation guidelines (11) (e.g., Inducer is included in INDIRECT-UPREGULATOR), and the instances of the relations on DrugBank that cannot be mapped by the dictionary are filtered out. As a result, 400,867 were used for training.

TABLE II. Micro-averaged F-scores on development set and test set. The results on the test set are shown only for the five submitted models. Bold is the best F-score.

Method type	Method	Development			Test		
		P	R	F1	P	R	F1
with database information	BioBERT-Large	0.788	0.746	0.766	-	-	-
	+desc	0.770	0.773	0.772	-	-	-
	+struct	0.759	0.784	0.771	-	-	-
	1-desc_struct	0.772	0.778	0.775	0.749	0.777	**0.763**
with distant supervised data	PubMedBERT	0.776	0.751	0.763	-	-	-
	4-ds_pretrain	0.766	0.726	0.746	0.752	0.739	0.746
	5-ds_pretrain_init	0.789	0.739	0.763	0.720	0.721	0.721
ensemble	**2-ds_desc_struct**	0.791	0.761	**0.776**	0.767	0.755	0.761
	3-ds_init_desc_struct	0.780	0.752	0.766	0.765	0.746	0.756

TABLE III. F-scores per class on development set.

Development	**1-de._st.**	**2-ds_de._st.**	**3-in._de._st.**	**4-ds_pr.**	**5-ds_pr._in.**
ACTIVATOR	0.754	0.766	0.748	0.728	0.748
AGONIST	0.770	0.783	0.785	0.769	0.780
AGONIST-ACTIVATOR	0.000	0.000	0.000	0.571	0.000
AGONIST-INHIBITOR	0.000	0.000	0.000	0.667	0.000
ANTAGONIST	0.915	0.931	0.916	0.916	0.925
DIRECT-REGULATOR	0.658	0.638	0.613	0.583	0.620
INDIRECT-DOWNREGULATOR	0.758	0.772	0.742	0.747	0.778
INDIRECT-UPREGULATOR	0.775	0.780	0.741	0.691	0.761
INHIBITOR	0.859	0.850	0.854	0.844	0.843
PART-OF	0.733	0.730	0.748	0.681	0.703
PRODUCT-OF	0.602	0.637	0.603	0.549	0.611
SUBSTRATE	0.728	0.726	0.713	0.690	0.703
SUBSTRATE_PRODUCT-OF	0.000	0.000	0.000	0.000	0.000
macro-average	0.580	0.585	0.574	0.649	0.575
micro-average	0.775	0.776	0.766	0.746	0.763

2) Denoising distant supervision data

Since the distant supervision data is automatically generated from a database, it contains a lot of false positives, which can be noise in training. To alleviate this, we built a denoising model that recognizes and reduces noise, assuming that negative examples of the DrugProt dataset have characteristics similar to noise.

The input to the denoising model is a sentence in which the target entities are masked. BERT is used as the sentence encoder. The BERT [CLS] token is passed to a single fully connected layer for binary classification. The parameters are updated by minimizing the cross-entropy loss.

The training of the denoising model is performed in a semi-supervised way by combining a part of the training data from the DrugProt dataset and distant supervision data. Three-way cross-validation is used to create three denoising models, where two-third of the training data is utilized for training the model with distant supervision data, and the remaining data is utilized for the validation. The detailed training flow of one model is as follows: We first prepare the train part of the DrugProt dataset as the training data. We then perform one epoch training of the denoising model on the training data. Next, we apply the trained denoising model to the distant supervision data and add the top 500 positive and negative examples to the training data. After training the model, we evaluate the performance of the binary classifier on the validation data set. Finally, we repeat training and adding distant supervision data until the performance of the validation data stops increasing.

We use the three denoising models to predict and denoise distant supervision data. We take the average of the prediction scores of the three models as the prediction score of the entire denoising model, and denoise distant supervision data by a threshold.

Data classified as non-noise with a score above a specified threshold (0.999 in the experiment) are added to the training data for relation extraction as positive instances, and data classified to be noise with a score above the threshold are added to the training data for relation extraction as negative instances.

3) Learning models using both distant and direct supervision data

We use the DrugProt dataset and denoised distant supervision data to train a BERT-based relation classification model. The input is a sentence with the target entity masked. BERT is used as the sentence encoder. The BERT [CLS] token is passed to a single fully connected layer for multiclass classification. The parameters are updated by minimizing the cross-entropy loss.

First, we train the relation extraction model on the denoised distant supervision data. Next, we initialize the weights of the model with the weights of the model trained on the distant supervision data, and train the model on the DrugProt dataset.

D. Ensemble

We combine the two models explained in section III *B* and section III *C* using an ensemble technique. We perform the ensemble by averaging the probability vectors of the two models after passing through the softmax layer.

IV. Experiments

A. Models

We have submitted the following five models.

1-desc_struct A model using the description and structure information of protein/drug entity.

2-ds_desc_struct The ensemble of models 1 and 4

3-ds_init_desc_struct The ensemble of models 1 and 5

4-ds_pretrain A model using distant supervision data. All parameters are pre-trained on distant supervision data

5-ds_pretrain_init A model using distant supervision data. Layers other than the fully connected layer are initialized with parameters pre-trained on distant supervision data

B. Experimental settings

Our system was implemented in Python3 (12), using Pytorch (13) as the machine learning library. For the model **1-desc_struct**, BioBERT-Large (14) is used as the text encoder. Adam (15) is used as the optimization method, and the model was trained with a batch size of 128. The BioBERT-Large was prepared separately for the input sentence and the entity description, and they are fine-tuned during training. The maximum sentence length was set to 128 for both the input sentence and the entity description.

For drugs, string matching was performed for the entry names, synonyms, product names, and brand names in DrugBank. For proteins, string matching was performed for the entry names, recommended names, alternative names, and gene names in Uniprot. As a result, 94% and 99% of drug and protein mentions in the training data set matched the DrugBank and Uniprot entries. When the entity could not link to database entries or the description and structure information is not registered in the database, we use an empty string as the input of BERT and CNNs, i.e., all tokens are replaced with the padding token. In the structural information encoding using character CNNs, the maximum sequence length of SMILES was set to 200, and that of amino acid sequences was set to 1,500. For both drugs and proteins, the character embedding dimension size d_c was set to 100, the convolution output vector dimension size was set to 16, and the convolution window size was set to [3,5,7]. Since we used three convolution windows, the dimension sizes of the structural vectors $\boldsymbol{h}_{struct1}$ and $\boldsymbol{h}_{struct2}$ are both 16×3=48.

For **4-ds_pretrain** and **5-ds_pretrain_init**, PubMedBERT (16) is used as the text encoder. Adam is used as the optimization method, and the model was trained with a batch size of 32. Optuna (17) is used to adjust the learning rate, weight decay, and dropout rate of hyperparameters. During the semi-supervised training of the denoiser, the top 500 distant supervision data in each epoch were added to the training data. In addition, a classification score of 0.999 was set as the threshold for denoiser classification.

V. Results

We evaluated the performance of proposed models on the development set and test set in Table II. Regarding the method using the description and structure information of the database, the F-score is improved in the development data set in both the case where the description information and the structure information is used individually, compared with the baseline BioBERT-Large model. Furthermore, the model **1-desc_struct**, which uses both description and structure information, further improved the F-score from the baseline model.

Comparing PubMedBERT to **5-ds_pretrain_init**, the F-score did not change much, and no performance improvement was obtained by utilizing the distant supervision data. On the contrary, **4-ds_pretrain** showed a significant decrease in performance. Table III shows the F-score for each interaction type. For most of the classes of F-scores and a micro average, **5-ds_pretrain_init** showed better performance than **4-ds_pretrain**. On the other hand, for AGONIST-ACTIVATOR and AGONIST-INHIBITOR with less training data, **4-ds_pretrain**, which initializes all weights pre-trained on distant supervision data, showed higher performance.

As for the ensembled models, The model **2-ds_desc_struct**, the ensemble of **1-desc_struct** and **4-ds_pretrain**, showed slightly higher performance than the model **1-desc_struct** and showed the highest F-score on the development data set. However, the performance of the model of the ensemble is slightly degraded from the original model.

VI. Conclusion

We propose two BERT-based drug-protein interaction extraction methods, utilizing entities description and structure information, and constructing distant supervision data set for more effective model training. We show that the model with constructed distant supervision data set does not improve overall performance, but improves F-scores for some interaction labels. The model utilizing entity description and structure information shows higher performance from the baseline model, and we think these results show the importance of considering various information about entities for the drug-protein interaction extraction task.

As future work, we would like to investigate the effective approach to construct and utilize distant supervision data set. In addition, we will discover useful database information other than description and structural information for drug-protein interaction extraction.

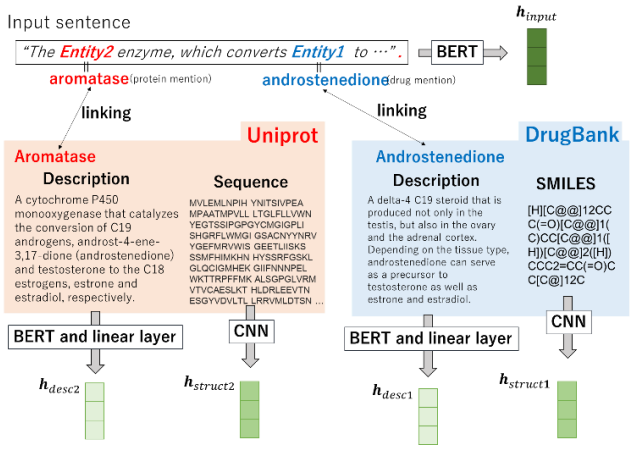

Fig. **1**. The model with description and structure information

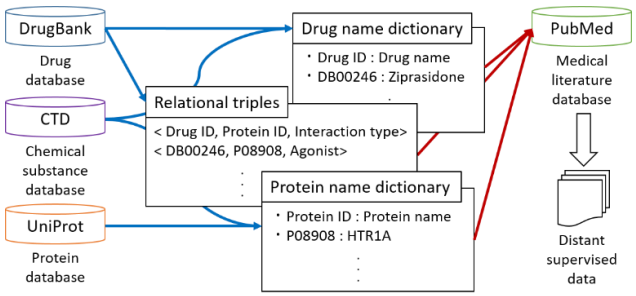

Fig. **2**. The flow of the creation of the distant supervision data

REFERENCES

1. Krallinger, Martin et al. "Overview of the BioCreative VI chemical-protein interaction Track." (2017).

2. Antonio Miranda et al., "Overview of DrugProt BioCreative VII track: quality evaluation and large scale text mining of drug-gene/protein relations." In Proceedings of the seventh BioCreative challenge evaluation workshop. 2021.

3. Mark Neumann et al. "ScispaCy: Fast and RobustModels for Biomedical Natural Language Process-ing". In: (Aug. 2019), pp. 319–327.doi:10.18653/v1 / W19 - 5034. eprint:arXiv : 1902 . 07669.url:https://www.aclweb.org/anthology/W19-5034.

4. Masaki Asada et al."Using drug descriptions and molecular structuresfor drug–drug interaction extraction from literature".In:Bioinformatics37.12 (Oct. 2020), pp. 1739–1746.issn: 1367-4803.doi:10 . 1093 / bioinformatics /btaa907. eprint:https : / / academic . oup . com /bioinformatics / article - pdf / 37 / 12 / 1739 /39119268/btaa907.pdf.url:https://doi.org/10.1093/bioinformatics/btaa907.

5. Wishart DS et al. "DrugBank 5.0: a major update tothe DrugBank database for 2018". In:Nucleic AcidsRes.(Jan. 2018).doi:10.1093/nar/gkx1037.

6. The UniProt Consortium. "UniProt: the universalprotein knowledgebase in 2021". In:Nucleic Acids Re-search49.D1 (Nov. 2020), pp. D480–D489.issn: 0305-1048.doi:10.1093/nar/gkaa1100. eprint:https://academic.oup.com/nar/article-pdf/49/D1/D480/35364103/gkaa1100.pdf.url:https://doi.org/10.1093/nar/gkaa1100

6. David Weininger. "SMILES, a chemical language andinformation system. 1. Introduction to methodologyand encoding rules". In:Journal of chemical informa-tion and computer sciences28.1 (1988), pp. 31–36.

8. Thien Huu Nguyen and Ralph Grishman. RelationExtraction: Perspective from Convolutional NeuralNetworks. In:Proceedings of the 1st Workshop onVector Space Modeling for Natural Language Process-ing. Denver, Colorado: Association for ComputationalLinguistics, June 2015, pp. 39–48.doi:10.3115/v1/W15-1506.url:https://aclanthology.org/W15-1506.

9. Allan Peter Davis et al. "Comparative Toxicoge-nomics Database (CTD): update 2021". In:NucleicAcids Research49.D1 (Oct. 2020), pp. D1138–D1143.issn: 0305-1048.doi:10.1093/nar/gkaa891. eprint:https://academic.oup.com/nar/article-pdf/49/D1/D1138/35364751/gkaa891.pdf.url:https://doi.org/10.1093/nar/gkaa891.

10. NCBI Resource Coordinators. "Database resources ofthe National Center for Biotechnology Information".In:Nucleic Acids Res.(Jan. 2016).doi:10.1093/nar/gkv1290.

11. Rabal, Obdulia, López, Jose Antonio, Lagreid, Astrid, & Krallinger, Martin. (2021). DrugProt corpus relation annotation guidelines [ChemProt - Biocreative VI]. https://doi.org/10.5281/zenodo.4957138 2021.

12. Van Rossum G, Drake FL. Python 3 Reference Manual. Scotts Valley, CA: CreateSpace; 2009.

13. Paszke Adam et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Advances in Neural Information Processing Systems 32 [Internet]. Curran Associates, Inc.; 2019. p. 8024–35. Available from: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

14. Jinhyuk Lee et al. "BioBERT: a pre-trained biomedi-cal language representation model for biomedical textmining". In:Bioinformatics36.4 (2020), pp. 1234–1240.

15. Diederik P. Kingma and Jimmy Ba. Adam: A Methodfor Stochastic Optimization. In:3rd InternationalConference on Learning Representations, ICLR 2015,San Diego, CA, USA, May 7-9, 2015, ConferenceTrack Proceedings. Ed. by Yoshua Bengio and YannLeCun. 2015.url:http://arxiv.org/abs/1412.6980.

16. Yu Gu et al. "Domain-Specific Language Model Pre-training for Biomedical Natural Language Process-ing". In: (2020). eprint:arXiv:2007.15779.

17. Takuya Akiba et al. Optuna: A Next-generation Hy-perparameter Optimization Framework. In:Proceed-ings of the 25rd ACM SIGKDD International Confer-ence on Knowledge Discovery and Data Mining. 2019.