

Toyota Technological Institute Ph.D. Thesis

Integrating Heterogeneous Domain Information
into Relation Extraction: A Case Study on
Drug-Drug Interaction Extraction

December 2022

Masaki Asada

Computational Intelligence Laboratory

Abstract

Relation extraction from the literature is a crucial task in natural language processing. It is very important to extract relationships between named entities from news articles and academic papers in various fields and to summarize these relations into structured data. However, the number of digital documents on the Internet is increasing daily, and it is impossible to perform relation extraction for all of the texts manually. Therefore, there is a need to study on automatic relation extraction from the literature using machine learning.

The development of deep neural networks has improved representation learning in various domains, including textual, graph structural, and relational triple representations. This development opened the door to new relation extraction beyond the traditional text-oriented relation extraction. However, research on the effectiveness of considering multiple heterogeneous domain information simultaneously is still under exploration, and if a model can take an advantage of integrating heterogeneous information, it is expected to exhibit a significant contribution to many problems in the world.

This thesis works on Drug-Drug Interactions (DDIs) from the literature as a case study and realizes relation extraction utilizing heterogeneous domain information. Drugs are related to information about various aspects, including textual information, information of molecular structures, categorical information, and related protein information. Therefore, extracting DDIs from the literature is a suitable target to verify the effectiveness of considering heterogeneous domain information in the relation extraction task.

First, a deep neural relation extraction model is prepared and its attention mechanism is analyzed. Next, a method to combine the drug molecular structure information and drug description information to the input sentence information is proposed, and the effectiveness of utilizing drug molecular structures and drug descriptions for the relation extraction task is shown.

Then, in order to further exploit the heterogeneous information, drug-related items, such as protein entries, medical terms and pathways are collected from multiple existing databases and a new data set in the form of a knowledge graph (KG) is constructed. A link prediction task on the constructed data set is conducted to obtain embedding representations of drugs that contain the heterogeneous domain information.

Finally, a method that integrates the input sentence information and the heterogeneous KG information is proposed. The proposed model is trained and evaluated on a widely used data set, and as a result, it is shown that utilizing heterogeneous domain information significantly improves the performance of relation extraction from the literature. Overall, this study demonstrates the importance of considering heterogeneous domain items in the information extraction task beyond the text-oriented information extraction.

Acknowledgments

First, and most of all, I would like to express my gratitude to my supervisor, Prof. Yutaka Sasaki. I have been indebted to him for more than five years, and it is impossible to get to this point without his continual support. He gave me sincere guidance for my research plan, for writing papers, and for preparing presentation slides.

I would also like to thank Assoc. Prof. Makoto Miwa, who has given me a lot of extremely important advice in my research on neural relation extraction. He is a great motivator in my decision to pursue a Ph.D. degree. During our daily discussions, he gave me very thoughtful comments that made my Ph.D. research even greater.

I would like to thank my co-supervisor, Prof. Kazuo Hotate for his great advice in the introduction part of my research presentation. I would like also to thank the members of my Ph.D. thesis review committee. I would like to thank Prof. Yukihiro Motoyama for giving me insightful comments about drugs. I would like to thank Prof. Norimichi Ukita for the great discussions about deep learning methods that helps me to improve my Ph.D. thesis. I would like to thank Prof. Yoshimasa Tsuruoka for his extremely insightful comments on natural language processing.

Another big thank you goes to all of my colleagues at Toyota Technological Institute, who have supported me. As a senior member of the Ph.D. program, Tomoki Tsujimura provided me with a lot of advice on my research. We had a lot of fun conversations outside of work.

My deepest gratitude goes to my family for their endless love and support during all these years. I would like to thank my parents for supporting my decisions with generosity.

Contents

1	Introduction	1
1.1	Motivations and Objectives	1
1.2	Contributions	3
1.3	Thesis Structure	4
2	New Approach to Relation Extraction: Unified Use of Heterogeneous Domain Information	6
2.1	Relation Extraction: An Overview	6
2.1.1	Task Definition	6
2.1.2	Data Sets and Corpora	6
2.1.3	Evaluation Metrics	8
2.1.4	Conventional Approaches Dealing with Relation Extraction from the Literature	9
2.2	Integrating Heterogeneous Domain Information into Relation Extraction .	11
3	Relation Extraction from Texts with Attention CNNs	13
3.1	Background	13
3.2	Preprocessing	14
3.3	Methods	14
3.3.1	Base CNN Model	15
3.3.2	Embedding Layer	16
3.3.3	Convolution Layer	16
3.3.4	Pooling Layer	17
3.3.5	Prediction Layer	17
3.3.6	Attention Mechanism	18
3.3.7	Training Method	19

3.4	Experimental Settings	19
3.4.1	DDI Data Sets	19
3.4.2	Initializing Word Embeddings	21
3.4.3	Hyper-parameter Tuning	21
3.5	Results and Discussions	22
3.5.1	Performance Analysis	22
3.5.2	Comparison with Existing Models	24
3.5.3	Comparison of Attention Mechanisms	24
3.5.4	Visual Analysis	25
3.6	Summary	25
4	Relation Extraction with a Single Kind of Domain Information	26
4.1	Background	26
4.2	Methods	26
4.2.1	Text-based DDI Extraction	26
4.2.2	Molecular Structure-based DDI Classification	27
4.2.3	DDI Extraction from Texts Using Molecular Structures	29
4.3	Experimental Settings	29
4.3.1	Data and Task for Molecular Structures	30
4.3.2	Training Settings	30
4.4	Results and Discussions	31
4.5	Summary	32
5	Relation Extraction with Multiple Domain Information	33
5.1	Background	33
5.2	Methods	34
5.2.1	Input Sentence Representation	35
5.2.2	Drug Description Representation	36

5.2.3	Molecular Structure Representation	36
5.2.4	DDI Extraction Using Database Information	38
5.2.5	Training	38
5.2.6	Ensemble	38
5.3	Experimental Settings	39
5.3.1	DrugBank Preprocessing	39
5.3.2	Drug Mention Linking	39
5.3.3	Training Settings	41
5.4	Results and Discussions	42
5.4.1	Pre-training of GNNs and CNNs on DrugBank	45
5.4.2	Can DrugBank Information Alone Extract DDIs from Texts?	47
5.4.3	Error Analysis	47
5.5	BioCreativeVII Track-1 DrugProt	48
5.5.1	Introduction	48
5.5.2	Task Definition	48
5.5.3	Methods	49
5.5.4	Experiments	52
5.6	Summary	54
6	Representing Heterogeneous Knowledge Graph	56
6.1	Background: An Overview of Knowledge Graph Representation	56
6.2	Heterogeneous Pharmaceutical Knowledge Graph with Textual Information	58
6.2.1	Constructing Heterogeneous Pharmaceutical Knowledge Graph . . .	59
6.2.2	Textual Information of Knowledge Graph	63
6.3	Learning Knowledge Graph Embeddings	64
6.3.1	Knowledge Graph Definition	64
6.3.2	Scoring Functions	64

6.3.3	Negative Sampling and Loss Functions	66
6.4	Methods	66
6.4.1	Initializing Node Embeddings	67
6.4.2	Aligning Entity Embeddings and Textual Embeddings	68
6.4.3	Augmenting KG Embeddings	68
6.5	Experimental Settings	69
6.5.1	Constructing Heterogeneous KG with Textual Information	69
6.5.2	Encoding Text Information	69
6.5.3	KG Embedding Training Settings	70
6.5.4	Task Setting	71
6.5.5	Hyper-parameter Settings	71
6.5.6	Implementation Details	72
6.6	Results and Discussions	72
6.6.1	Analysis of the Data Ombalance of the Constructed KG	74
6.6.2	Ablation Study of Augmentation Method	76
6.6.3	Effect of Node Type Filtering	76
6.6.4	Case Study	77
6.7	Summary	78
7	Integrating Heterogeneous Domain Information for Relation Extraction	80
7.1	Background	80
7.2	Method	81
7.2.1	Obtaining Heterogeneous KG Embeddings	81
7.2.2	DDI Extraction from Texts with Heterogeneous KG Embeddings . .	82
7.3	Experimental Settings	85
7.3.1	Mention Linking	85
7.3.2	Link Prediction Settings	85

7.3.3	DDI Extraction Settings	86
7.4	Results and Discussions	86
7.4.1	Selecting Score Functions	90
7.4.2	Ablation Study on Model Architecture	90
7.4.3	Ablation Study on Heterogeneous KG Node Types	92
7.4.4	Verification of DDI Label Leakage from KGs	92
7.4.5	Learning Curve	93
7.4.6	Analysis of Prediction Results	94
7.5	Summary	95
8	Conclusions	97
8.1	Summary	97
8.2	Future Work	98
8.2.1	Employing the Deep Neural Entity Linking Method	98
8.2.2	Joint Learning of BERT and KG Embeddings	98
8.2.3	End-to-End DDI Extraction	99
8.2.4	Extension to Other Tasks	99

List of Tables

2.1	Contingency table of true positives (TP), false positives (FP), true negatives (TN) and false negative (FN) for binary DDI classification	8
3.1	An example of the preprocessed input sentence	15
3.2	Statistics for the DDIExtraction-2013 shared task data set	21
3.3	Hyper-paramters	22
3.4	Statistics of the development data set	22
3.5	Performance of softmax/ranking CNN models with and without the attention mechanism	23
3.6	Comparison with existing models	23
3.7	Comparison of attention mechanisms on CNN models with ranking objective function	24
4.1	Evaluation on DDI extraction from texts	31
4.2	F-scores of each DDI type	31
4.3	Accuracy of binary classification on DrugBank pairs	31
4.4	Classification of DDIs in texts by molecular structure-based DDI classification model	31
5.1	Hyper-parameters for CNNs	41
5.2	Hyper-parameters for GNNs	42
5.3	Evaluation on DDI extraction from texts on the test set	44
5.4	Evaluation on DDI extraction from texts on the development set	44
5.5	Performance on individual DDI types in F-scores	45
5.6	Individual F-scores on 5-fold cross-validated training data set	46
5.7	Comparisons of F-scores on different parts of the test set	47
5.8	Accuracy of binary classification on the DrugBank pairs	47

5.9	Evaluation on DDI extraction from texts with or without pre-training of GNNs for the molecular structure and CNNs for the description	48
5.10	Micro-averaged F-scores on DrugProt development set and test set	52
5.11	F-scores per class on DrugProt development set	53
6.1	Statistics of heterogeneous pharmaceutical KG entities	60
6.2	Statistics of heterogeneous pharmaceutical KG edges for each relation type	61
6.3	The percentage of nodes that have each type of text	70
6.4	Comparison of MRR performance for each method	73
6.5	Summary of the best settings for each relation	75
6.6	Ablation study of text information on Augmentation method (ComplEx score function)	76
6.7	Comparison of averaged MRR performance for w/ (with) and w/o (without) entity type filtering	77
6.8	The content of the text in the examples where the difference between the rank of textual model and the rank of non-textual model is largest for each relation type	79
7.1	Hyper-parameters for link prediction and DDI extraction model	87
7.2	The comparison of link prediction performance on heterogeneous KG . . .	88
7.3	The comparison of DDI extraction performance on DDIExtraction-2013 test data set	89
7.4	The comparison of F-scores for individual DDI types and macro-averaged F-score on DDIExtraction-2013 test data set	90
7.5	The comparison of DDI extraction performances with different score functions for training KG embeddings	90
7.6	The ablation study on model architecture	91
7.7	The ablation study on node types	92

7.8	The DDI extraction performance for drug pairs that are included in the constructed heterogeneous KG and drug pairs that are not included in KG	93
7.9	Case studies of the proposed model	96

List of Figures

1.1	Yearly number of papers with the MeSH category of " <i>Chemicals and Drugs</i> " in PubMed	2
1.2	Heterogeneous domain information about a mention in the literature	3
2.1	The visualization of some instances in the DDIExtraction-2013 data set	7
2.2	The outline of integrating heterogeneous domain information into the re- lation extraction	12
3.1	The overview of CNN with attention mechanism	15
3.2	The visualization of attention weights	25
4.1	Overview of the CNN-based DDI extraction model that use molecular structure information	29
5.1	Overview of the DDI extraction model with drug descriptions and drug molecular structures	35
5.2	Illustration of molecular fingerprints. This figure shows the extraction of several fingerprint subgraphs from a molecular structure when radius is 2.	37
5.3	Linking between mentions and DrugBank entry	40
5.4	F-scores for different sentence lengths on the 5-fold cross-validated training data set	49
5.5	The model with description and structure information	50
6.1	Illustration of the Heterogeneous Pharmaceutical Knowledge Graph	59
6.2	Overview of methods: (A) Initializing node embeddings (Initialization), (B) Aligning entity embeddings and textual embeddings (Alignment), and (C) Augmenting KG embeddings (Augmentation)	67
6.3	The distribution of the frequency of nodes in the train data set	75
7.1	Overview of the levitated marker	81
7.2	Heterogeneous KG with additional drug molecular structure information	82

7.3	The DDI extraction model which utilize heterogeneous information of drugs	84
7.4	Learning curve of baseline model and the proposed model on 5-fold cross-validation data sets	94
7.5	The confusion matrices (left: baseline, right: proposed method)	95

1 Introduction

1.1 Motivations and Objectives

Relation extraction from the literature is a crucial task in natural language processing. It is very important to extract relationships between named entities from the literature and summarize these relations into structured data. However, the number of digital documents on the Internet is increasing daily, and it is impossible to conduct relation extraction for all of the texts manually. Figure 1.1 shows the yearly number of papers in the medicine and biomedical journal literature search engine PubMed. The number of papers is generally increasing year by year, and there is a need to study on automatic relation extraction from the literature using machine learning.

It was considered very difficult to utilize different kinds of information at the same time. However, with the advent of deep learning [1], it has become possible to represent each attribute as fixed-size numerical vectors using methods such as word2vec [2] and node2vec [3]. As for text representation, the advent of Bidirectional Encoder Representations from Transformer (BERT) [4] has made it more powerful. Although these developments have opened the door to new relation extraction beyond the traditional text-oriented relation extraction, the research on simultaneously considering multiple heterogeneous domain information is still under exploration. The author believes that the new techniques will provide clues for the integrated use of heterogeneous domain information. This thesis elaborates to establish deep learning-based methods that can comprehensively handle heterogeneous information relevant to mentions in relation extraction from the literature.

Similar to this motivation is multimodal learning [5] and multi-view learning [6]. Multimodal learning is a method that simultaneously considers information in two or three different modalities, such as text and image, or text and audio. Multi-view learning is an emerging direction in machine learning which considers learning with multiple views

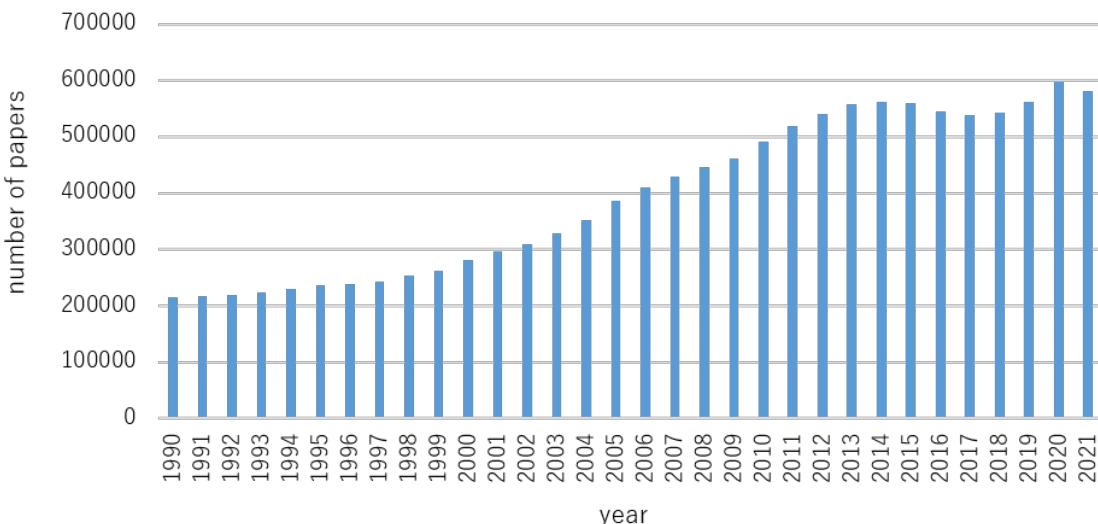


Figure 1.1: Yearly number of papers with the MeSH category of “*Chemicals and Drugs*” in the medicine and biomedical journal literature search engine PubMed. The number of papers is generally increasing year by year.

to improve the generalization performance. Multi-view learning is also known as data fusion or data integration from multiple feature sets. Figure 1.2 shows an example of the heterogeneous domain information of an entity in the literature. The examples of drug entities in the literature show multiple kinds of domain information related to drugs. Drugs contain various kinds of information, such as molecular structures, descriptions, and categorical hierarchies. This thesis further develops multimodal learning and incorporates multi-view learning to integrate a lot of domain information in the unified vector space, and utilizes them to conduct the relation extraction task.

One thing which must be clear here is that the relation extraction task consists of two parts: (i) recognition of named entities in input sentences and (ii) extraction of the relation between identified named mentions, and this thesis focuses only on the relation extraction part. This is because many heterogeneous factors need to be considered in order to extract relations from the literature.

Extracting relations between drugs from the literature is an appropriate problem in verifying the effectiveness in considering heterogeneous domain information. Medicines help us feel better and stay healthy. But sometimes drug interactions can cause problems.

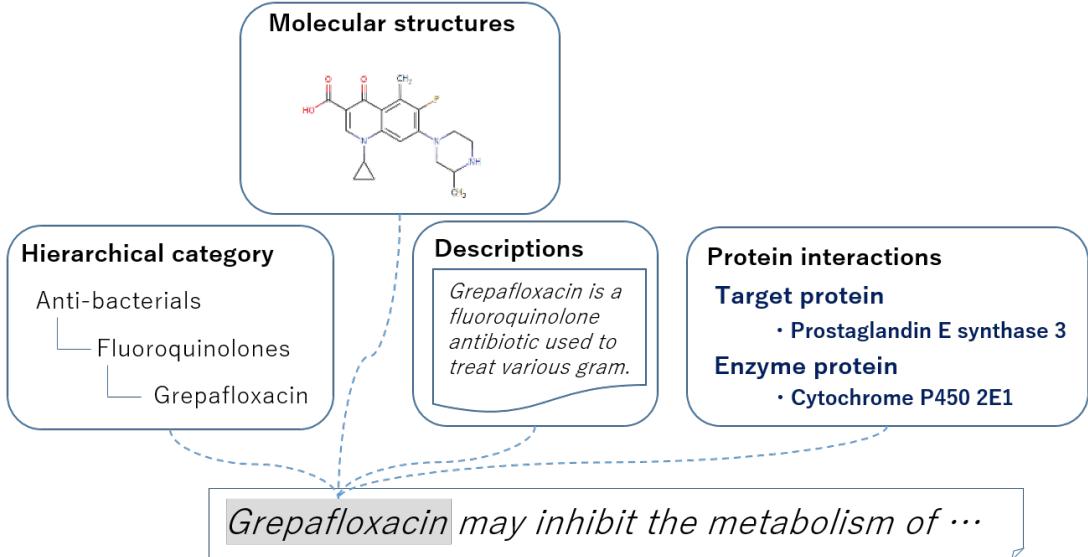


Figure 1.2: Heterogeneous domain information about a mention in the literature

Drug interactions can be classified into the following three broad categories [7]:

Drug-drug interaction A reaction between two (or more) drugs.

Drug-food interaction A reaction between a drug and a food or beverage.

Drug-condition interaction A reaction that occurs when taking a drug while having a certain medical condition.

This thesis chooses the drug-drug interaction (DDI) extraction task as a case study. When considering DDIs, it is necessary to refer to the descriptions in drug articles for achieving evidence-based medicine [8]; therefore it is important to extract DDIs from the literature. Hence, the author emphasizes that the goal is to determine whether a sentence in the articles represents that two drugs interact or not. It is not a goal to determine whether the two drugs actually interact. For instance, when a sentence just mentions a pair of drug entities without justifying their interaction, the model cannot extract the DDI of the two drugs in the context of the sentence.

1.2 Contributions

The contributions of this thesis are shown as follows:

- To utilize heterogeneous domain information in relation extraction.

- To construct a new benchmark data set in the form of knowledge graphs to use heterogeneous information.
- To propose a neural architecture that integrally uses sentence information and domain information.
- To propose methods to utilize the heterogeneous domain representation of drugs trained from the constructed data set to extract DDIs from texts.
- To evaluate the advantage of the proposed method using drug-drug interaction extraction as a case study.
- To show that several kinds of heterogeneous drug information are complementary and their effective combinations can largely improve the DDI extraction performance.
- To show that the proposed neural architecture can effectively utilize heterogeneous domain information for DDI extraction.

1.3 Thesis Structure

The remainder of this thesis is organized as follows.

Chapter 2 provides the novelties of the thesis based on an overview of relation extraction from the literature. First, how the relation extraction corpus was created and the relation extraction task settings, evaluation metrics and data set statistics are described. Then, the history of how machine learning-based automatic relation extraction from the literature is described. Finally, a new approach that integrates heterogeneous domain information into relation extraction tasks is introduced.

Chapter 3 provides a base neural model for relation extraction as a preliminary, and reviews the effectiveness of the method of employing an attention mechanism in the CNN-based model on relation extraction tasks. This model is a text-oriented relation extraction model and is not capable of domain information.

Chapter 4 reviews the study on relation extraction from texts considering molecular structures of drugs in addition to the context of input sentences. A baseline model with the word2vec algorithm and Convolutional Neural Networks (CNNs) is addressed in this section. Since this model cannot deal with heterogeneous domain information, the following chapters develop relation extraction models that can consider heterogeneous domain information.

Chapter 5 proposes a relation extraction model that can consider two types of domain information: the molecular structure information of the drug and the description information of the drug in a case study. In addition, the relation extraction model is applied to the drug-protein interaction (DPI) extraction task which was held in the BioCreative VII shared task. This section and later sections employ Bidirectional Encoder Representations from Transformer (BERT) [4] as the baseline model. This chapter includes work from the published paper [9].

Chapter 6 builds a novel data set in the form of a Knowledge Graph (KG) to consider further heterogeneous information about drugs. The link prediction task is conducted on the created heterogeneous KG and the embedding representations of the drugs are obtained. This chapter is based on the published work [10].

Chapter 7 utilizes the heterogeneous KG embeddings of the drugs for DDI extraction task. The effectiveness of combining the input sentence representation and KG representation is described. This chapter is based on [11].

Chapter 8 summarizes the findings in this thesis and concludes the thesis. Finally, the further developments of the model and the future plan are discussed.

2 New Approach to Relation Extraction: Unified Use of Heterogeneous Domain Information

This chapter firstly provides an overview of relation extraction tasks. Then a new approach that integrates heterogeneous domain information into relation extraction tasks is introduced.

2.1 Relation Extraction: An Overview

2.1.1 Task Definition

With the expanding use of the Internet, a large amount of the literature becomes available every day in the form of news articles, research publications, question answering forums, and social media. It is important to develop techniques for extracting information automatically from these documents and summarizing relations into structured data. The entity like the person and the organization is the most basic unit of information. Occurrences of entities in a sentence are often linked through well-defined relations; e.g., occurrences of person and organization in a sentence may be linked through relations such as *employed_at*. The task of relation extraction is to identify such relations automatically. [12]

Figure 2.1 shows some examples of annotated sentences. These sentences are included in the abstract of the PubMed article. The first sentence indicates that there are three entities in the sentence, and the entity *S-ketamine* and the entity *ticlopidine* have relation *Effect*. The aim of the relation extraction task is to classify the given all possible entity pairs into well-defined relations.

2.1.2 Data Sets and Corpora

The construction of a common data set is essential for the development of research of automatic relation extraction from the literature and several data sets have been created to assist the development of relation extraction methods. These data sets were constructed

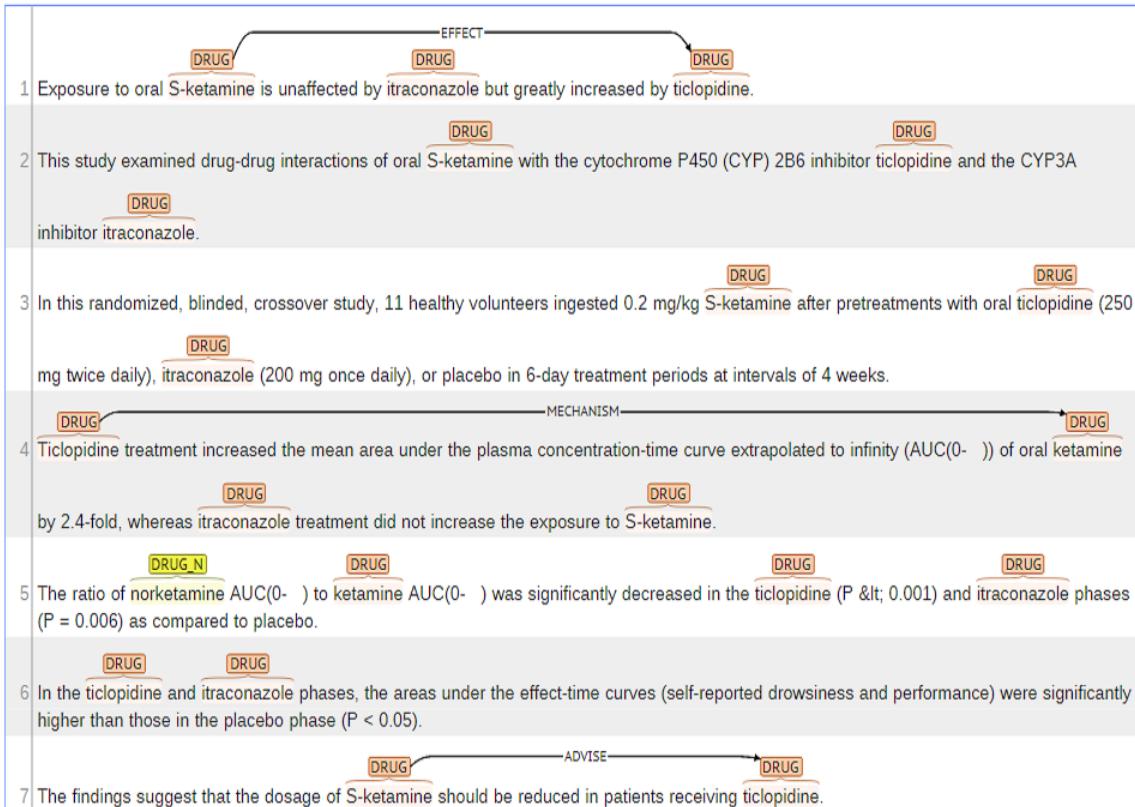


Figure 2.1: The visualization of some instances in the DDIExtraction-2013 data set

by trusted annotators who are experts in their fields.

For the relation extraction task in the general field, Automatic Content Extraction (ACE) [13] data set which contains named entities, relations, and events in various languages, mostly from news articles was constructed. Another widely used data set is CoNLL04 [14], which also contains entities and relations for the general field. SemEval-2010 task 8 [15] data set was created a few years later and used widely to develop relation extraction models.

For the biomedical field, protein-protein interaction was the first target for relation extraction task and several protein-protein interaction extraction data sets such as AIMed [16] and BioInfer [17] data sets were created. Drug-drug interaction (DDI) extraction has also been studied. DDIExtraction-2013 [18] data set was constructed and it is used as a standard benchmark data set. BioCreative VI ChemProt [19] data set targeted Chemical-Protein relations and is a large relation extraction data set with multiple fine-

grained relations. Later, BioCreative VII DrugProt [20] task data set was constructed, which contains 13 types of drug-protein interactions.

This thesis focuses on the DDIExtraction-2013 task and investigates the effectiveness of heterogeneous domain information about drugs.

2.1.3 Evaluation Metrics

Evaluation is relation-oriented and based on the standard Precision, Recall, and micro-averaged F-score metrics. Precision is also called Positive Predictive Value (PPV) and Recall is also called Sensitivity. *TP* (*True Positives*) correspond to the number of pairs correctly identified by the model as having a relation, *TN* (*True Negatives*) correspond to the total number of pairs correctly identified as not having a relation. *FP* (*False Positives*) correspond to the total number of missed pairs which have a relation and *FN* (*False Negatives*) correspond to the total number of missed pairs which do not have a relation. Table 2.1 shows the contingency table of TP, FP, TN and FN for a binary classification task.

		Ground Truth	
		Relation	No Relation
Prediction	Relation	TP	FP
	No Relation	FN	TN

Table 2.1: Contingency table of true positives (TP), false positives (FP), true negatives (TN) and false negative (FN) for binary DDI classification

Using these statistics, The evaluation metrics Precision (P), Recall (R) and F1-score (F1) are calculated as follows:

$$P = \frac{TP}{TP + FP} \quad (2.1)$$

$$R = \frac{TP}{TP + FN} \quad (2.2)$$

$$F1 = \frac{2PR}{P + R} \quad (2.3)$$

The above described statistics and metrics are typically used in the case of binary classification. However, the DDIExtraction-2013 data set has four types of positive labels. When the F-score is used as an evaluation metric for multi-class classification, there are

two types of F-scores, micro-averaged and macro-averaged F-scores. While micro-averaged F-score is calculated by constructing a global contingency table and then calculating precision and recall, macro-averaged F-score is calculated by first calculating precision and recall for each type and then taking the average of these results. The micro- and macro-averaged metrics for Precision, Recall and F-score are described in the following equations, where c indicates the interaction category.

$$P_{micro} = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FP_c}, \quad R_{micro} = \frac{\sum_c TP_c}{\sum_c TP_c + \sum_c FN_c} \quad (2.4)$$

$$F1_{micro} = \frac{2P_{micro}R_{micro}}{P_{micro} + R_{micro}} \quad (2.5)$$

$$P_{macro} = \frac{1}{|c|} \sum_c P_c, \quad R_{macro} = \frac{1}{|c|} \sum_c R_c \quad (2.6)$$

$$F1_{macro} = \frac{2P_{macro}R_{macro}}{P_{macro} + R_{macro}} \quad (2.7)$$

In this task, the micro-averaged F1-score is used as the main evaluation metric.

2.1.4 Conventional Approaches Dealing with Relation Extraction from the Literature

CNN-based Relation Extraction The first application of deep Convolutional Neural Networks (CNNs) to the relation extraction task was the CNNs model proposed by Zeng et al. [21]. CNN models take a candidate relation instance that is represented by the word embeddings and position embeddings as input. Then, the input is fed to the convolutional layer in order to extract features with filters of different sizes. Subsequently, the pooling layer performs down-sampling on the feature maps, which provides two advantages: firstly, it can identify the most relevant and essential local features; and secondly, it reduces the computational complexity which the reduced resolution. Finally, for classifying the relation type, the feature vector generated by pooling is fed into a fully connected softmax layer. The experiments showed that this CNN-based method can capture the semantic information and relative position of words better than conventional Support Vector Machines (SVMs) [22] methods.

RNNs for Relation Extraction Recurrent Neural Networks (RNNs) mainly address sequential data. Each output depends on the previous values in this structure, which is similar to the human memory mechanism. Generally, these networks are composed of an input layer, a hidden layer that is connected to itself, and an output layer. In the standard formulation, RNNs suffer from exploding and vanishing gradient problems. To address these problems, Long Short-Term Memory (LSTM) [23] units and Gated Recurrent Units (GRU) [24] networks model the hidden layer with a memory cell that controls the extent of what is to be forgotten (and to be incorporated) given the input, the previous state, and the current state. These networks can explicitly learn when to forget information and when to store it, which is fit for an input of arbitrary length. Undoubtedly, these variants are efficient in the relation extraction task. As for RNN-based approaches, Socher et al. [25] presented a novel relation classification model that learns vectors in the syntactic tree path that connects two nominals to determine their semantic relationship.

Transformers and BERT Family for Relation Extraction Many neural network-based methods have been proposed since Liu et al. [26] first tackled the DDI extraction task with a neural network-based method. RNNs and CNNs have been widely used for NLP tasks, but both methods have the problem that it is difficult to capture relationships between distant words in a sentence. Transformer [27], a model that utilizes the attention mechanism, has been newly devised and has been believed to solve this problem. The attention is a mechanism for calculating the importance of each word in a sentence. Given a single input sentence, a contextualized word representation can be obtained by a self-attention mechanism that calculates the importance of each word in the sentence. While CNNs are unable to consider context outside the window size and RNNs are thought to be difficult to capture relationships between distant words, Transformers are thought to overcome these shortcomings.

Bidirectional Encoder Representation from Transformers (BERT) [4] is a Transformer

model that is pre-trained on a large unlabeled corpus. The approach of fine-tuning the pre-trained BERT shows high performance on various downstream tasks, including the relation extraction task.

2.2 Integrating Heterogeneous Domain Information into Relation Extraction

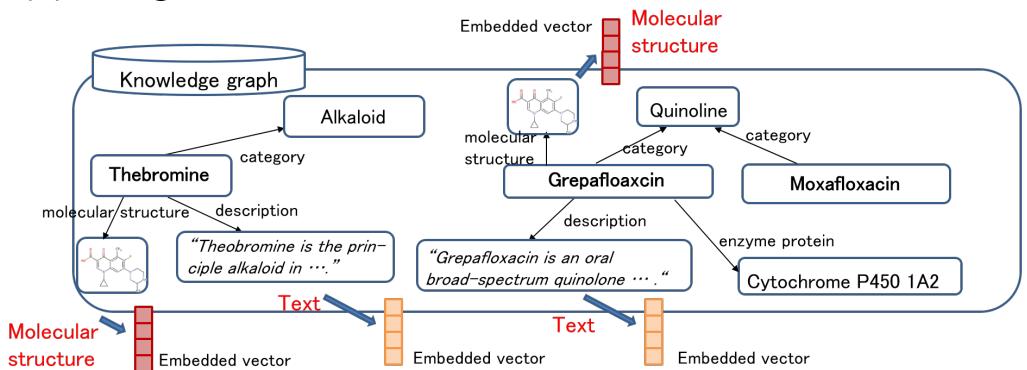
This section outlines the novel approach to integrating heterogeneous domain information into the relation extraction task. This thesis develops in three steps.

Using individual domain information Figure 2.2 (1) shows the first step of the proposed method. Here, the relation extraction that utilizes the drug-related information from external databases are addressed. The molecular structures and descriptions of drugs are represented as vectors separately by a neural network model. Details based on case studies are provided in Chapter 4 and Chapter 5.

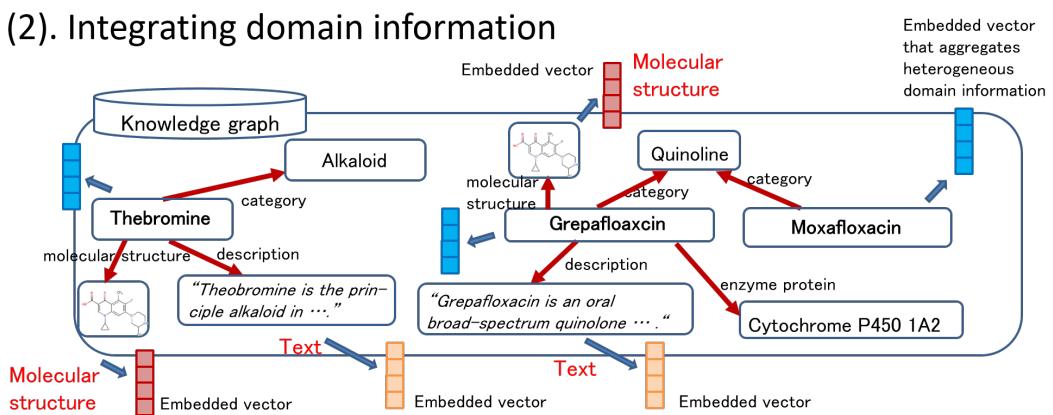
Integrating Heterogeneous domain information Figure 2.2 (2) shows the second step of the development. Heterogeneous drug-related knowledge, including the molecular structures, is represented in a unified vector space. Diverse information is integrated and the vectors that aggregate heterogeneous domain information are obtained.

Combining heterogeneous domain information and input sentences Figure 2.2 (3) displays the main novelty of this thesis. In this figure, the vectors of aggregated heterogeneous domain information are effectively utilized for relation extraction from the literature. The heterogeneous domain information is represented in a unified vector space. This approach enables to integrate heterogeneous knowledge in relation extraction.

(1). Using individual domain information



(2). Integrating domain information



(3). Combining heterogeneous domain information and input sentence

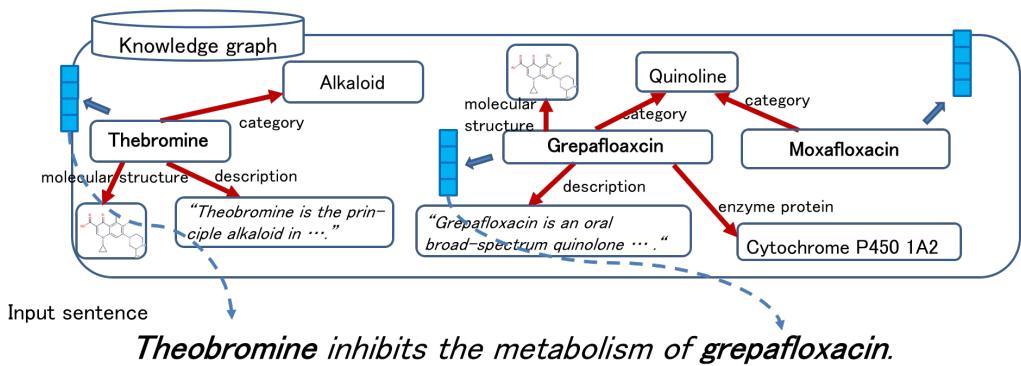


Figure 2.2: The outline of integrating heterogeneous domain information into the relation extraction

3 Relation Extraction from Texts with Attention CNNs

As a preliminary, this chapter introduces a baseline relation extraction model with an attention mechanism for a CNN-based relation extraction model [28] and reviews the usefulness of the attention mechanism. This verification is the preliminary experiment to a method that utilizes heterogeneous information. The technique of preprocessing data set and detailed statistics of the DDIExtraction-2013 article, which are common in subsequent chapters in this thesis, are discussed.

The overview of this chapter is as follows:

- This chapter provides an attention mechanism that can boost the performance on CNN-based DDI extraction.
- The DDI extraction model with the attention mechanism achieves the performance with an F-score of 69.12%, which is competitive with other state-of-the-art DDI extraction models when the performance without negative instance filtering [29] is compared.

3.1 Background

For the DDI extraction, deep neural network-based methods have recently drawn a considerable attention [26, 30, 31]. Deep neural networks have been widely used in the NLP field. They show high performance on several NLP tasks without requiring manual feature engineering. CNNs and RNNs are often employed for network architectures. Among these, CNNs have the advantage that they can be easily parallelized and the calculation is thus fast with recent Graphical Processing Units (GPUs).

Liu et al. [26] showed that a CNN-based model can achieve a high accuracy on the task of DDI extraction. Sahu et al. [31] proposed an RNN-based model with the attention mechanism to tackle the DDI extraction task and show state-of-the-art performance. The

integration of an attention mechanism into a CNN-based relation extraction is proposed by Wang et al. [32]. This is applied to a general domain relation extraction task SemEval-2010 Task 8 [15]. Their model showed state-of-the-art performance on the task. CNNs with attention mechanisms, however, are not evaluated on the task of DDI extraction.

This chapter reviews a attention mechanism that is integrated into a CNN-based DDI extraction model). The attention mechanism extends the attention mechanism by [32] in that it deals with anonymized entities separately from other words and incorporates a smoothing parameter. A CNN-based relation extraction model is implemented and the mechanism into the model is integrated. The model is evaluated on the DDIEtraction-2013 data set [18].

3.2 Preprocessing

When three or more drug mentions appear in an input sentence, the sentence is duplicated 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.

According to the settings of many existing methods [26, 31, 33], before tokenizing an input sentence, the mentions of the target drugs in the pair are replaced with DRUG1 and DRUG2 according to their order of appearance. The other mentions of drugs are replaced with DRUGOTHER.

Table 3.1 shows an example of preprocessing when the input sentence *Exposure to oral S-ketamine is unaffected by itraconazole but greatly increased by ticlopidine* is given with a target entity pair. By performing preprocessing, it is possible to prevent the DDI extraction model to be specialized for the surface forms of the drugs in a training data set and to perform DDI extraction using the information of the whole context.

3.3 Methods

An attention mechanism for a CNN-based DDI extraction model is described. The overview of the DDI extraction model is illustrated in Figure 3.1. The model extracts

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

Table 3.1: An example of preprocessing on the sentence *Exposure to oral S-ketamine is unaffected by itraconazole but greatly increased by ticlopidine* for each target pair.

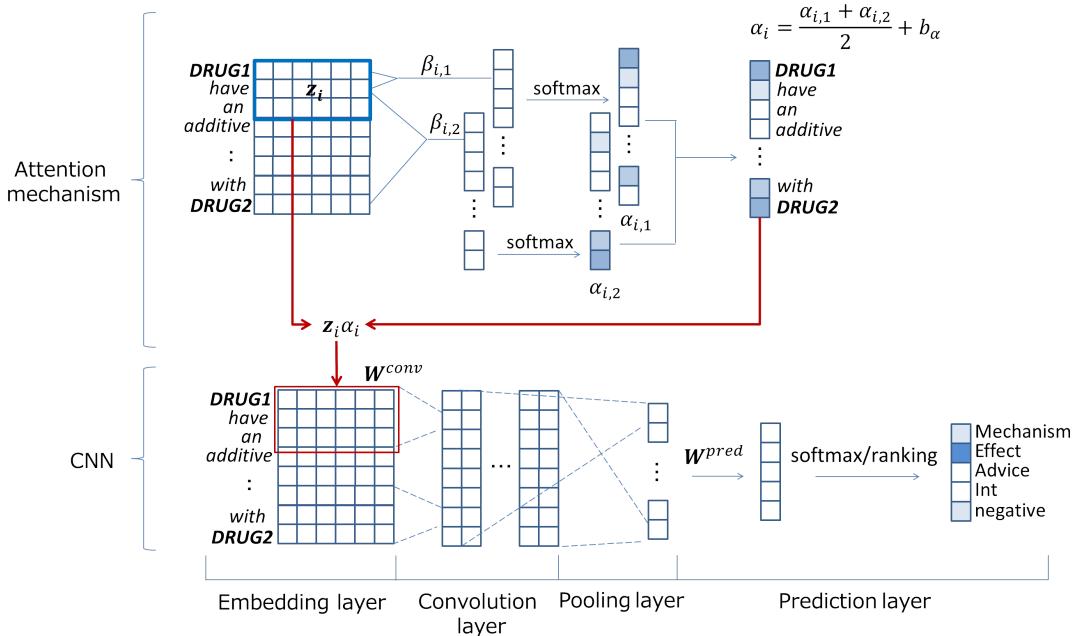


Figure 3.1: The overview of CNN with attention mechanism

interactions from sentences with drugs are given. This section first presents preprocessing of input sentences, then introduces the base CNN model and explains the attention mechanism. Finally, the training method is described.

3.3.1 Base CNN Model

The CNN model for extracting DDIs is based on Zeng et al. [21]. In addition to the original objective function, the ranking-based objective function by dos-Santos et al. [34] is employed. The model consists of four layers: embedding, convolution, pooling, and prediction layers. The CNN model is shown at the lower half of Figure 3.1.

3.3.2 Embedding Layer

In the embedding layer, each word in the input sentence is mapped to a real-valued vector representation using an embedding matrix that is initialized with pre-trained embeddings.

Given an input sentence $S = (w_1, \dots, w_n)$ with drug entities e_1 and e_2 , each word w_i is first converted into a real-valued vector \mathbf{w}_i^w by an embedding matrix $\mathbf{W}^{emb} \in \mathbb{R}^{d_w \times |V|}$ as follows:

$$\mathbf{w}_i^w = \mathbf{W}^{emb} \mathbf{v}_i^w, \quad (3.1)$$

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

d_{wp} -dimensional word position embeddings $\mathbf{w}_{i,1}^p$ and $\mathbf{w}_{i,2}^p$ that correspond to the relative positions from first and second target entities are prepared, respectively. The word embedding \mathbf{w}_i^w and these word position embeddings $\mathbf{w}_{i,1}^p$ and $\mathbf{w}_{i,2}^p$ as in the following Equation (3.2) are concatenated, and the resulting vector is used as the input to the subsequent convolution layer:

$$\mathbf{w}_i = [\mathbf{w}_i^w; \mathbf{w}_{i,1}^p; \mathbf{w}_{i,2}^p]. \quad (3.2)$$

3.3.3 Convolution Layer

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

$$\mathbf{z}_{i,k} = [\mathbf{w}_{[i-(k-1)/2]}^T; \dots; \mathbf{w}_{[i-(k+1)/2]}^T]^T. \quad (3.3)$$

The convolution to the embedding matrix is applied as follows:

$$m_{i,j,k} = f(\mathbf{W}_{k,j}^{conv} \odot \mathbf{z}_{i,k} + b), \quad (3.4)$$

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

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3.5)$$

3.3.4 Pooling Layer

The max pooling [35] is employed to convert the output of each filter in the convolution layer into a fixed-size vector as follows:

$$\mathbf{c}_k = [c_{1,k}, \dots, c_{d_c,k}], \quad c_{j,k} = \max_i m_{i,j,k}. \quad (3.6)$$

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

$$\mathbf{c} = [\mathbf{c}_{k_1}; \dots; \mathbf{c}_{k_i}; \dots; \mathbf{c}_{k_K}]. \quad (3.7)$$

3.3.5 Prediction Layer

The relation types are predicted using the output of the pooling layer. First \mathbf{c} is converted into scores using a weight matrix $\mathbf{W}^{pred} \in \mathbb{R}^{o \times d_p}$:

$$\mathbf{s} = \mathbf{W}^{pred} \mathbf{c}, \quad (3.8)$$

where o is the total number of relationships to be classified and $\mathbf{s} = [s_1, \dots, s_o]$. Then the following two different objective functions are employed for prediction.

Softmax \mathbf{s} is converted into the probability of possible relations \mathbf{p} by a softmax function:

$$\mathbf{p} = [p_1, \dots, p_o], \quad p_j = \frac{\exp(s_j)}{\sum_{l=1}^o \exp(s_l)}. \quad (3.9)$$

The cross-entropy loss function L_{CE} is defined as in the Equation (3.10) when the gold type distribution \mathbf{y} is given. \mathbf{y} is a one-hot vector where the probability of the gold label is 1 and the others are 0.

$$L_{CE} = - \sum \mathbf{y} \log \mathbf{p} \quad (3.10)$$

Ranking The ranking-based objective function is employed following [34]. Using the scores \mathbf{s} in Equation (3.8), the loss is calculated as follows:

$$\begin{aligned} L_{ranking} &= \log(1 + \exp(\gamma(m^+ - s_y))) \\ &\quad + \log(1 + \exp(\gamma(m^- + s_c))), \end{aligned} \quad (3.11)$$

where m^+ and m^- are margins, γ is a scaling factor, y is a gold label, and c ($\neq y$) is a negative label with the highest score in \mathbf{s} . γ to 2, m^+ is set to 2.5 and m^- is set to 0.5 following [34].

3.3.6 Attention Mechanism

The attention mechanism is based on the input attention by Wang et al. [32]¹. The attention mechanism is different from the base one in which separate attentions are prepared for entities and a bias term is incorporated to adjust the smoothness of attentions. The attention mechanism is illustrated at the upper half of Figure 3.1.

The word indexes of the first and second target drug entities in the sentence are defined as e_1 and e_2 , respectively. The set of indices are denoted as $E = \{e_1, e_2\}$ and the indexes of the entities as $j \in \{1, 2\}$. The attentions are calculated using:

$$\beta_{i,j} = \mathbf{w}_{e_j} \cdot \mathbf{w}_i \quad (3.12)$$

$$\alpha_{i,j} = \begin{cases} \frac{\exp(\beta_{i,j})}{\sum_{1 \leq l \leq n, l \notin E} \exp(\beta_{l,j})}, & \text{if } i \notin E \\ a_{drug}, & \text{otherwise} \end{cases} \quad (3.13)$$

$$\alpha_i = \frac{\alpha_{i,1} + \alpha_{i,2}}{2} + b_\alpha. \quad (3.14)$$

¹The attention-based pooling in [32] is not incorporated. This is left for future work.

Here, a_{drug} is an attention parameter for entities and b_α is the bias term. a_{drug} and b_α are tuned during training. If E is set to empty and b_α is set to zero, the attention will be the same as one by Wang et al. [32]. The attentions α_i are incorporated into the CNN model by replacing Equation (3.4) with the following equation:

$$m_{i,j,k} = f(\mathbf{W}_j^{conv} \odot \mathbf{z}_{i,k} \alpha_i + b). \quad (3.15)$$

3.3.7 Training Method

L2 regularization [36] is used to avoid over-fitting. The following objective functions L'_* ($L'_{softmax}$ or $L'_{ranking}$) is used by incorporating the L2 regularization on weights to Equation (5.11).

$$\begin{aligned} L'_* = L_* + \lambda(&||\mathbf{W}^{emb}||_F^2 + ||\mathbf{W}^{conv}||_F^2 \\ &+ ||\mathbf{W}^{pred}||_F^2) \end{aligned} \quad (3.16)$$

Here, λ is a regularization parameter and $|| \cdot ||_F$ denotes the Frobenius norm. All the parameters including the weights \mathbf{W}^{emb} , \mathbf{W}^{conv} , and \mathbf{W}^{pred} , biases b and b_α , and the attention parameter a_{drug} are updated to minimize L'_* . The adaptive moment estimation (Adam) [37] is used for the optimizer. Training data set is randomly shuffled and divided into mini-batch samples in each epoch.

3.4 Experimental Settings

3.4.1 DDI Data Sets

The DDIExtraction-2011 [38] workshop (First Challenge Task on Drug-Drug Interaction Extraction) focuses on the extraction of DDIs from biomedical texts and aims to promote the development of text mining and information extraction systems applied to the pharmacological domain in order to reduce time spent by the medical experts reviewing the literature for potential DDIs. For performance comparisons between machine learning models, it is important to evaluate models on the data set created by trusted annotators with shared problem settings. The construction of a common data set is essential

for the development of research on automatic DDI extraction from the literature. The main goal of this shared task is to have a benchmark for the comparison of advanced techniques, rather than competitive aspects. The DDIEtraction-2013 [18] follows up the DDIEtraction-2011 whose main goal was the detection of DDIs from texts. The DDIEtraction-2013 includes the classification of DDI types in addition to DDI detection. Figure 2.1 shows some examples of labeled sentences. In the DDIEtraction-2013 task, it is necessary not only to detect DDIs but also to correctly classify DDIs into the type of the relations such as “Effect” and “Mechanism”. Additionally, while the data set used for the DDIEtraction-2011 task was composed by texts describing DDIs from the DrugBank [39], the new data set for DDIEtraction-2013 also includes MEDLINE abstracts in order to deal with different types of texts and language styles.

The aim of the DDIEtraction-2013 task is to classify a given pair of drugs into the following four interaction types or no interaction:

Mechanism : A sentence describes pharmacokinetic mechanisms of a DDI,

e.g., *Grepafloxacine may inhibit the metabolism of theobromine.*

Effect : A sentence represents the effect of a DDI, e.g., *Methionine may protect against the ototoxic effects of gentamicin.*

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

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

The DDIEtraction-2013 task relies on the DDI corpus, which is a semantically annotated corpus of documents describing DDIs from the DrugBank database and MEDLINE abstracts on the subject of DDIs. The statistics of the DDIEtraction-2013 task data

	Train		Test	
	DrugBank	MEDLINE	DrugBank	MEDLINE
#documents	572	142	158	33
#sentences	5,675	1,301	973	326
#pairs	26,005	1,787	5,265	451
#positive DDIs	3,789	232	884	95
#negative DDIs	22,216	1,555	4,381	356
#Mechanism pairs	1,257	62	278	24
#Effect pairs	1,535	152	298	62
#Advice pairs	818	8	214	7
#Int pairs	179	10	94	2

Table 3.2: Statistics for the DDIExtraction-2013 shared task data set

set are shown in Table 3.2. As shown in this table, the number of pairs that have no interaction (negative pairs) is larger than that of pairs that have interactions (positive pairs).

3.4.2 Initializing Word Embeddings

Skip-gram [2] was employed for the pre-training of word embeddings. The 2014 MEDLINE/PubMed baseline distribution is used, and the size of vocabulary was 1,630,978. The embedding of the drugs, i.e., *DRUG1*, *DRUG2* and *DRUGOTHER* are initialized with the pre-trained embedding of the word *drug*. The embeddings of training words that did not appear in the pre-trained embeddings, as well as the word position embeddings, are initialized with the random values drawn from a uniform distribution and normalized to unit vectors. Words whose frequencies are one in the training data were replaced with an *UNK* word during training, and the embedding of words in the test data set that did not appear in both training and pre-trained embeddings were set to the embedding of the *UNK* word.

3.4.3 Hyper-parameter Tuning

The official training data set is split into two parts: training and development data sets. The hyper-parameters are tuned on the development data set on the softmax model without attentions. Table 3.3 shows the best hyper-parameters on the softmax model without attentions. The same hyper-parameters are applied to the other models. The

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

Table 3.3: Hyper-paramters

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

Table 3.4: Statistics of the development data set

statistics of the development data set is shown in Table 3.4. The sizes of the convolution windows is set to [3, 4, 5] that are the same as in Kim et al. [40]. The word position embedding size is chosen from {10, 20, 30, 40, 50}, the convolutional filter size from {10, 50, 100, 200}, the learning rate of Adam from {0.01, 0.001, 0.0001}, the mini-batch size from {10, 20, 50, 100, 200}, and the L2 regularization parameter λ from {0.01, 0.001, 0.0001, 0.00001}.

3.5 Results and Discussions

This section first summarizes the performance of the models and compares the performance with existing models. Then attention mechanisms are compared and finally some results for the analysis of the attentions are illustrated.

3.5.1 Performance Analysis

The performance of the base CNN models with two objective functions, as well as with or without the attention mechanism, is summarized in Table 3.5. The incorporation of the attention mechanism improved the F-scores by about 2 percent points (pp) on models with both objective functions. Both improvements were statistically significant ($p < 0.01$) with t -test. This shows that the attention mechanism is effective for both models. The

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

Table 3.5: Performance of softmax/ranking CNN models with and without the attention mechanism. The highest scores are shown in bold.

Methods	P (%)	R (%)	F (%)
No negative instance filtering			
CNN [26]	75.29	60.37	67.01
MCCNN [41]	-	-	67.80
SCNN [30]	68.5	61.0	64.5
Joint AB-LSTM [31]	71.82	66.90	69.27
Attention model	76.30	63.25	69.12
With negative instance filtering			
FBK-irst [29]	64.6	65.6	65.1
Kim et al. [42]	-	-	67.0
CNN [26]	75.72	64.66	69.75
MCCNN [41]	75.99	65.25	70.21
SCNN [30]	72.5	65.1	68.6
Joint AB-LSTM [31]	73.41	69.66	71.48

Table 3.6: Comparison with existing models

improvement of F-scores from the least performing model (softmax objective function without the attention mechanism) to the best performing model (ranking objective func-

	<i>P</i> (%)	<i>R</i> (%)	<i>F</i> (%)
No attention	74.79 (± 2.41)	60.99 (± 2.65)	67.10 (± 1.09)
Input attention by Wang et al. [32]	73.48 (± 1.96)	59.58 (± 1.51)	65.77 (± 0.80)
Attention	76.30 (± 2.66)	63.25 (± 2.59)	69.12 (± 0.71)
Attention w/o separate attentions a_{drug}	74.03 (± 2.11)	63.30 (± 2.41)	68.17 (± 0.71)
Attention w/o the bias term b_α	71.56 (± 2.18)	64.19 (± 2.21)	67.62 (± 0.96)

Table 3.7: Comparison of attention mechanisms on CNN models with ranking objective function

tion with the attention mechanism) is 3.19 pp (69.12% versus 65.93%), and this shows both objective function and attention mechanism are key to improve the performance. When looking into the individual types, ranking function with the attention mechanism archived the best F-scores on *Mechanism*, *Effect*, *Advice*, while the base CNN model achieved the best F-score on *Int*.

3.5.2 Comparison with Existing Models

A comparison with the existing state-of-the-art models is shown in Table 3.6. The performance is mainly compared without negative instance filtering, which omits some apparent negative instance pairs with rules [29], since it is not incorporated. The performance of the existing models with negative instance filtering is also shown for reference.

In the comparison without negative instance filtering, the model outperformed the existing CNN models [26, 30, 41]. The model was competitive with Joint AB-LSTM model [31] that was composed of multiple RNN models.

3.5.3 Comparison of Attention Mechanisms

The attention mechanism is compared with the input attention of Wang et al. [32] to show the effectiveness of the attention mechanism. Table 3.7 shows the comparison of the attention mechanisms. The base CNN-based model with ranking loss is also shown for reference, and the results of ablation tests. As is shown in the table, the attention mechanism by Wang et al. [32] did not work in DDI extraction. However, the attention improved the performance. This result shows that the extensions are crucial for modeling attentions in DDI extraction. The ablation test results show that both extensions to the attention mechanism, i.e., separate attentions for entities and incorporation of the bias

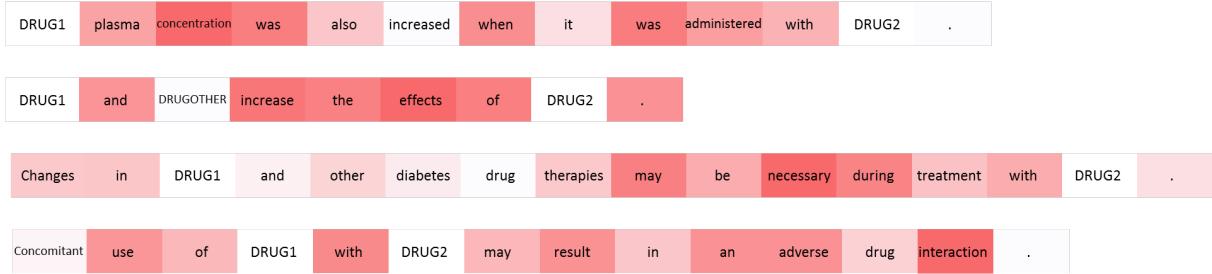


Figure 3.2: The visualization of attention weights. The dark part indicates that the attention value is large.

term, are effective for the task.

3.5.4 Visual Analysis

Figure 3.2 shows a visualization of attentions on some sentences with DDI pairs using the attention mechanism. In the first sentence, *DRUG1* and *DRUG2* have a *Mechanism* interaction. The attention mechanism successfully highlights the keyword *concentration*. In the second sentence, which have an *Effect* interaction, the attention mechanism put high weights on *increase* and *effects*. The word *necessary* has a high weight on the third sentence with an *Advice* interaction. For an *Int* interaction in the last sentence, the word *interaction* is most highlighted.

3.6 Summary

In this chapter, an attention mechanism for relation extraction is provided as a preliminary. Base CNN-based DDI extraction models is built with two different objective functions, softmax and ranking, and The attention mechanism is incorporated into the models. The performance on the DDIExtraction-2013 data set is evaluated, and it is shown that both the attention mechanism and ranking-based objective function are effective for extracting DDIs. The final model achieved an F-score of 69.12%. This model is a baseline, text-oriented relation extraction model and is not capable of domain information.

4 Relation Extraction with a Single Kind of Domain Information

This chapter reviews a study on dealing with a single kind of domain information, which is the molecular structure of drugs as the preparatory step in the case study of using heterogeneous information on drugs. A method that combines the input sentence information and molecular structure information is reviewed based on [43]. The summary of this chapter is three-fold:

- A neural method to extract DDIs from texts with the related molecular structure information is described.
- GCNs is applied to pairwise drug molecules and it is shown that GCNs can predict DDIs between drug molecular structures with high accuracy.
- It is shown that the molecular information is useful in extracting DDIs from texts.

4.1 Background

In parallel to the progress in DDI extraction from texts, Graph Convolutional Networks (GCNs) have been proposed and applied to estimate physical and chemical properties of molecular graphs such as solubility and toxicity [44–46].

It is a very challenging attempt to consider different kinds of items such as text information and molecular structure information at the same time as described in Chapters 5-7. However, both the input sentence vector encoded by CNNs and the molecular structure vector encoded by GCNs are embedded in a low-dimensional real-valued vector space, so both information is effectively utilized for the DDI extraction task.

4.2 Methods

4.2.1 Text-based DDI Extraction

The proposed model for extracting DDIs from texts is based on the CNN model by Zeng et al. [21]. When an input sentence $S = (w_1, w_2, \dots, w_N)$ is given, word embedding \mathbf{w}_i^w of

w_i and word position embeddings $\mathbf{w}_{i,1}^p$ and $\mathbf{w}_{i,2}^p$ that correspond to the relative positions from the first and second target entities are prepared, respectively. These embeddings are concatenated as in Equation (4.1), and the resulting vector are used as the input to the subsequent convolution layer:

$$\mathbf{w}_i = [\mathbf{w}_i^w; \mathbf{w}_{i,1}^p; \mathbf{w}_{i,2}^p], \quad (4.1)$$

where $[;]$ denotes the concatenation. The expression for each filter j with the window size k_l is calculated as:

$$\mathbf{z}_{i,l} = [\mathbf{w}_{i-(k_l-1)/2}, \dots, \mathbf{w}_{i-(k_l+1)/2}], \quad (4.2)$$

$$m_{i,j,l} = \text{relu}(\mathbf{W}_j^{conv} \odot \mathbf{z}_{i,l} + b^{conv}), \quad (4.3)$$

$$m_{j,l} = \max_i m_{i,j,l}, \quad (4.4)$$

where L is the number of windows, \mathbf{W}_j^{conv} and b^{conv} are the weight and bias of CNN, and max indicates max pooling [35].

The output of the convolution layer is converted into a fixed-size vector that represents a textual pair as follows:

$$\mathbf{m}_l = [m_{1,l}, \dots, m_{J,l}], \quad (4.5)$$

$$\mathbf{h}_t = [\mathbf{m}_1; \dots; \mathbf{m}_L], \quad (4.6)$$

where J is the number of filters.

Prediction $\hat{\mathbf{y}}_t$ is obtained by the following fully connected neural networks:

$$\mathbf{h}_t^{(1)} = \text{relu}(\mathbf{W}_t^{(1)} \mathbf{h}_t + \mathbf{b}_t^{(1)}), \quad (4.7)$$

$$\hat{\mathbf{y}}_t = \text{softmax}(\mathbf{W}_t^{(2)} \mathbf{h}_t^{(1)} + \mathbf{b}_t^{(2)}), \quad (4.8)$$

where $\mathbf{W}_t^{(1)}$ and $\mathbf{W}_t^{(2)}$ are weights and $\mathbf{b}_t^{(1)}$ and $\mathbf{b}_t^{(2)}$ are bias terms.

4.2.2 Molecular Structure-based DDI Classification

Drug pairs are represented in molecular graph structures using two GCN methods: CNNs for fingerprints (NFP) [44] and Gated Graph Neural Networks (GGNN) [45]. They both

convert a drug molecule graph G into a fixed size vector \mathbf{h}_g by aggregating the representation \mathbf{h}_v^T of an atom node v in G . Atoms are represented as nodes and bonds are represented as edges in the graph.

NFP first obtains the representation \mathbf{h}_v^t by the following equations [44].

$$\mathbf{m}_v^{t+1} = \mathbf{h}_v^t + \sum_{w \in N(v)} \mathbf{h}_w^t, \quad (4.9)$$

$$\mathbf{h}_v^{t+1} = \sigma(\mathbf{H}_t^{deg(v)} \mathbf{m}_v^{t+1}), \quad (4.10)$$

where \mathbf{h}_v^t is the representation of v in the t -th step, $N(v)$ is the neighbors of v , and $\mathbf{H}_t^{deg(v)}$ is a weight parameter. \mathbf{h}_v^0 is initialized by the *atom features* of v . $deg(v)$ is the degree of a node v and σ is a sigmoid function. NFP then acquires the representation of the graph structure

$$\mathbf{h}_g = \sum_{v,t} \text{softmax}(\mathbf{W}^t \mathbf{h}_v^t), \quad (4.11)$$

where \mathbf{W}^t is a weight matrix.

GGNN first obtains the representation \mathbf{h}_v^t by using Gated Recurrent Unit (GRU)-based recurrent neural networks [45] as follows:

$$\mathbf{m}_v^{t+1} = \sum_{w \in N(v)} \mathbf{A}_{e_{vw}} \mathbf{h}_w^t \quad (4.12)$$

$$\mathbf{h}_v^{t+1} = \text{GRU}([\mathbf{h}_v^t; \mathbf{m}_v^{t+1}]), \quad (4.13)$$

where $\mathbf{A}_{e_{vw}}$ is a weight for the *bond type* of each edge e_{vw} . GGNN then acquires the representation of the graph structure.

$$\mathbf{h}_g = \sum_v \sigma(i([\mathbf{h}_v^T; \mathbf{h}_v^0])) \odot (j(\mathbf{h}_v^T)), \quad (4.14)$$

where i and j are linear layers and \odot is the element-wise product.

The representation of a molecular pair is obtained by concatenating the molecular graph representations of drugs g_1 and g_2 , i.e., $\mathbf{h}_m = [\mathbf{h}_{g_1}; \mathbf{h}_{g_2}]$.

Prediction $\hat{\mathbf{y}}_m$ is obtained as follows:

$$\mathbf{h}_m^{(1)} = \text{relu}(\mathbf{W}_m^{(1)} \mathbf{h}_m + \mathbf{b}_m^{(1)}), \quad (4.15)$$

$$\hat{\mathbf{y}}_m = \text{softmax}(\mathbf{W}_m^{(2)} \mathbf{h}_m^{(1)} + \mathbf{b}_m^{(2)}), \quad (4.16)$$

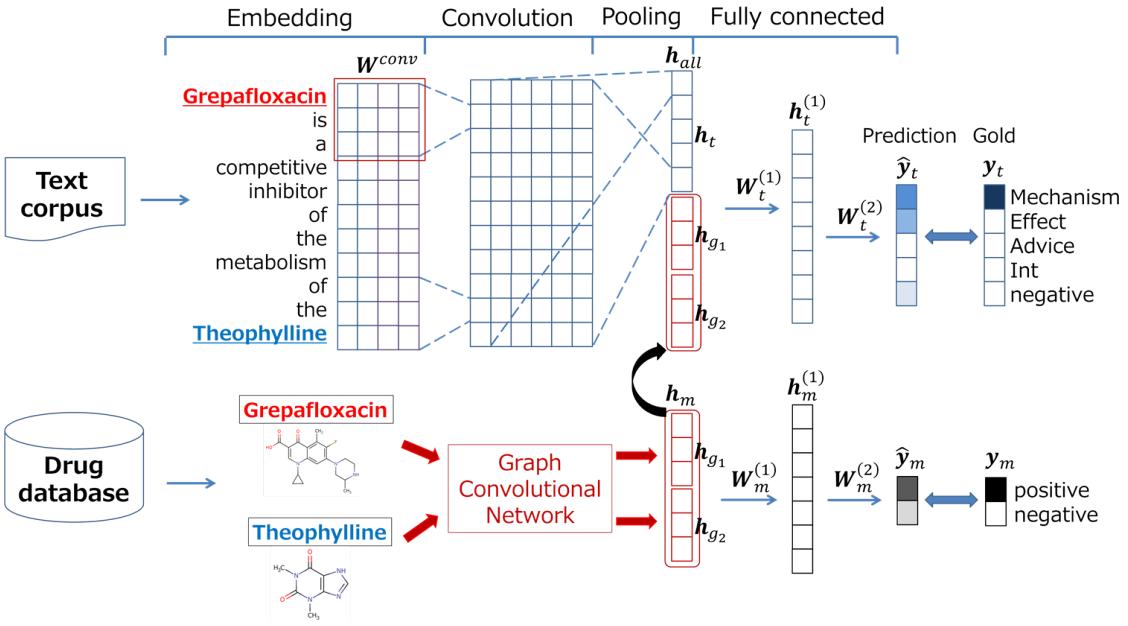


Figure 4.1: Overview of the CNN-based DDI extraction model that use molecular structure information

where $\mathbf{W}_m^{(1)}$ and $\mathbf{W}_m^{(2)}$ are weights and $\mathbf{b}_m^{(1)}$ and $\mathbf{b}_m^{(2)}$ are bias terms.

4.2.3 DDI Extraction from Texts Using Molecular Structures

The simultaneous use of textual and molecular information is achieved by concatenating the text-based and molecule-based vectors: $\mathbf{h}_{all} = [\mathbf{h}_t; \mathbf{h}_m]$. Molecule-based vectors are normalized. Then \mathbf{h}_{all} is used instead of \mathbf{h}_t in Equation 4.7.

In training, the molecular-based DDI classification model is trained. The molecular-based classification is conducted by minimizing the loss function $L_m = -\sum \mathbf{y}_m \log \hat{\mathbf{y}}_m$. The parameters for GCNs are fixed and the text-based DDI extraction model is trained by minimizing the loss function. $L_t = -\sum \mathbf{y}_t \log \hat{\mathbf{y}}_t$.

4.3 Experimental Settings

As preprocessing, sentences are split into words using the GENIA tagger [47]. The drug mentions of the target pair are replaced with *DRUG1* and *DRUG2* according to their order of appearance. Other drug mentions are also replaced with *DRUGOTHER*. Negative instance filtering is not employed unlike other existing methods, e.g., Liu et al. [26], since the focus is to evaluate the effect of the molecular information on texts.

Mentions in texts are linked to DrugBank entries by string matching. The mentions and the names are lowercased in the entries and chose the entries with the most overlaps. As a result, 92.15% and 93.09% of drug mentions in train and test data set matched the DrugBank entries.

4.3.1 Data and Task for Molecular Structures

255,229 interacting (positive) pairs are extracted from DrugBank. Note that, unlike text-based interactions, DrugBank only contains the information about interacting pairs; there are no detailed labels and no information for non-interacting (negative) pairs. Thus the same number of pseudo-negative pairs are generated by randomly pairing drugs and removing those in positive pairs. To avoid overestimation of the performance, drug pairs mentioned in the test set of the text corpus are deleted. Positive and negative pairs are split into 4:1 for training and test data, and the molecular GCN-based model is evaluated on the classification accuracy.

To obtain the graph of a drug molecule, the SMILES [48] string encoding of the molecule is taken as input from DrugBank and then converted into the graph using RD-Kit [49]. For the *atom features*, randomly embedded vectors are used for each atom (i.e., C, O, N, ...). 4 *bond types*, single, double, triple, and aromatic, are used.

4.3.2 Training Settings

Mini-batch training is employed using the Adam optimizer [37]. L2 regularization is used to avoid over-fitting. The bias term $\mathbf{b}_t^{(2)}$ is tuned for negative examples in the final softmax layer. Pre-trained word embeddings trained by using the word2vec tool [2] on the 2014 MEDLINE/PubMed baseline distribution are employed. The vocabulary size was 215,840. The embedding of the drugs, i.e., DRUG1 and DRUG2 were initialized with the pre-trained embedding of the word *drug*. The embeddings of training words that did not appear in the pre-trained embeddings were initialized with the average of all pre-trained word embeddings. Words that appeared only once in the training data were replaced with

Methods	P	R	F (%)
Liu et al. [26]	75.29	60.37	67.01
Zheng et al. [50]	75.9	68.7	71.5
Lim et al. [33]	74.4	69.3	71.7
Text-only	71.97	68.44	70.16
+ NFP	72.62	71.81	72.21
+ GGNN	73.31	71.81	72.55

Table 4.1: Evaluation on DDI extraction from texts

DDI Type	Mech.	Effect	Adv.	Int (%)
Text-only	69.52	69.27	79.81	48.18
+ NFP	72.70	72.44	79.56	46.98
+ GGNN	73.83	71.03	81.62	45.83

Table 4.2: F-scores of each DDI type

Methods	Accuracy (%)
NFP	94.19
GGNN	98.00

Table 4.3: Accuracy of binary classification on DrugBank pairs

Methods	P	R	F	Acc. (%)
NFP	15.56	48.93	23.61	45.78
GGNN	15.11	57.10	23.90	37.72

Table 4.4: Classification of DDIs in texts by molecular structure-based DDI classification model

an *UNK* word during training, and the embedding of words in the test data set that did not appear in both training and pre-trained embeddings were set to the embedding of the *UNK* word. Word position embeddings are initialized with random values drawn from a uniform distribution. The molecule-based vectors of unmatched entities are set to zero vectors.

4.4 Results and Discussions

Table 4.1 shows the performance of DDI extraction models. The performance without negative instance filtering or ensemble is shown for a fair comparison. The increase in recall and F-score are observed by using molecular information, which results in the state-of-the-art performance with GGNN.

Both GCNs improvements were statistically significant ($p < 0.05$ for NFP and $p < 0.005$ for GGNN) with randomized shuffled test [51].

Table 4.2 shows F-scores on individual DDI types. The molecular information improves

F-scores especially on type *Mechanism* and *Effect*.

The accuracy of binary classification on DrugBank pairs is also evaluated by using only the molecular information in Table 4.3. The performance is high, although the accuracy is evaluated on automatically generated negative instances.

Finally, the molecular-based DDI classification model trained on DrugBank is applied to the DDIExtraction 2013 task data set. Since the DrugBank has no detailed labels, all four types of interactions are mapped to positive interactions and evaluated the classification performance. The results in Table 4.4 show that GCNs produce higher recall than precision and the overall performance is low considering the high performance on DrugBank pairs. This might be because the interactions of drugs are not always mentioned in texts even if the drugs can interact with each other and because hedged DDI mentions are annotated as DDIs in the text data set. The DDI extraction model is trained only with molecular information by replacing \mathbf{h}_{all} with \mathbf{h}_m , but the F-scores were quite low (< 5%). These results show that textual relations cannot be predicted only with molecular information.

4.5 Summary

This chapter reviewed a neural method for relation extraction using both textual information and molecular structures. The model was evaluated on the DDI extraction task as a case study. The results show that DDIs can be predicted with using molecular structure information. Since this model cannot deal with heterogeneous domain information, the following chapters develop relation extraction models that can consider heterogeneous domain information.

5 Relation Extraction with Multiple Domain Information

This chapter includes work from the published paper Asada et al. (2021a) [9] and Iinuma et al. (2021) [52]. This chapter proposes a novel relation extraction method that utilizes two kinds of domain information. The DDI extraction task is chosen as a case study. A method to utilize the description and the structure of the entity obtained from drug database DrugBank as well as large-scale raw text information is proposed. DrugBank is in focus because the DDIEtraction 2013 shared task data set is created based on the DrugBank database. Other databases are left for future work. Specifically, the description and molecular structure information of drugs in the database are utilized. The information from large-scale raw texts is incorporated by using a Bidirectional Encoder Representations from Transformers (BERT) model [4] pre-trained on large-scale raw text.

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. It is shown that 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.

5.1 Background

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 a deep understanding of DDI interaction descriptions often requires domain knowledge of drugs. Various drug information, such as detailed descriptions and molecular structure information on drugs, are registered in drug databases. Furthermore, models pre-trained on large-scale raw text show significant improvements in various natural language processing (NLP) tasks [4].

Effective use of such external information is necessary to reduce the reliance on annotated text.

This chapter extends the previous section at the following points.

- The token representation changes from word2vec to contextualized vectors obtained by SciBERT. As a result, the performance of the baseline with the state-of-the-art performance is remarkably improved.
- The neural molecular GNN [53] that considers relatively large fragments of atoms and better represents molecular structures is employed.
- Drug descriptions registered in the drug database is used and it is shown that drug description information is useful for extracting DDIs from the corpus for some DDI types.
- It is found that 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.

5.2 Methods

The overview of the proposed method is illustrated in Figure 5.1. For the baseline model, the convolutional neural network (CNN)-based DDI extraction model [43] that receives an input sentence with a target drug pair and classifies the pair into a specific DDI type is employed. The input sentence is enriched using SciBERT [54], which is a BERT model trained on large-scale biomedical and computer science text. The drug description representation of the target drugs is obtained using SciBERT and the molecular structure representation of the target drugs using molecular graph neural network (GNN) model proposed by Tsubaki et al. [53]. These drug description and molecular structure representation are combined with the enriched input sentence representation and classify the target drug pair into a specific DDI type.

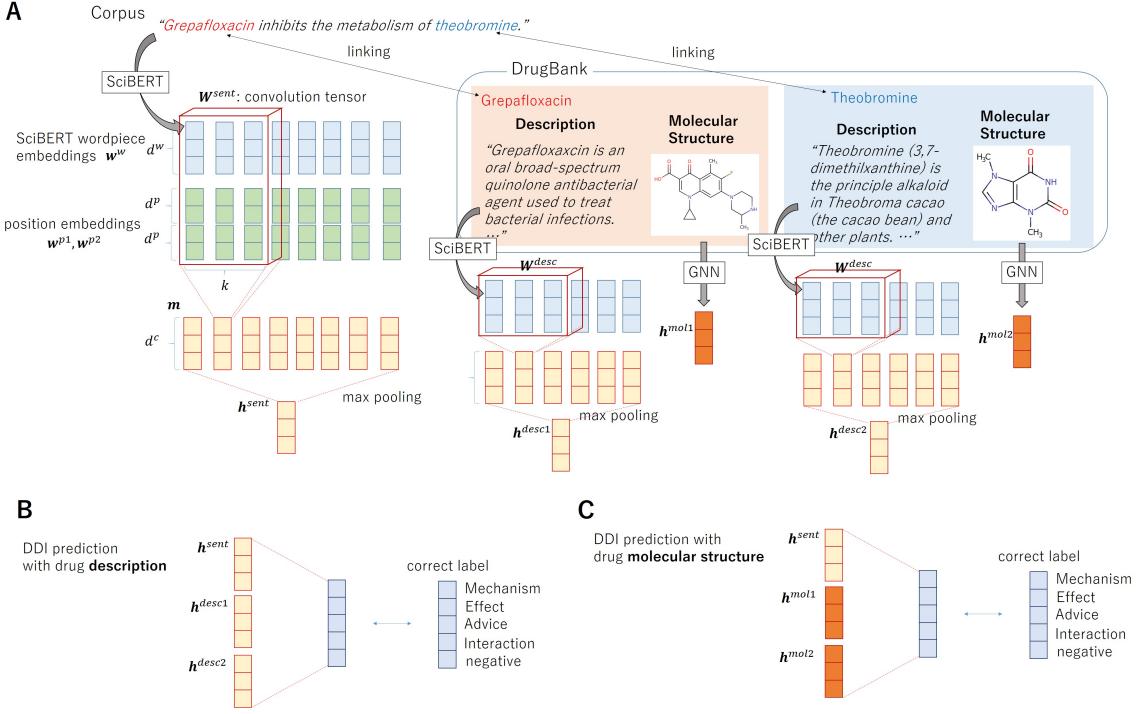


Figure 5.1: Overview of the DDI extraction model with drug descriptions and drug molecular structures. (A) Illustrates how to encode input sentences, drug descriptions and drug molecular structures. (B) and (C) show the prediction layer when the drug description representation and the drug molecular structure representation are used.

5.2.1 Input Sentence Representation

This section follows the previous section to preprocess the input sentences. Then a pre-processed input sentence is converted into a real-valued fixed size vector by BERT and CNN-based model [4, 21] and shows the model in the left part of Figure 5.1A. Given an input sentence $S = (w_1, \dots, w_n)$ with drug mentions m_1 and m_2 , the sentence is first split into wordpieces (a.k.a., subwords) by the WordPiece algorithm [55]. Each wordpiece w_i is converted into a real-valued pre-trained contextualized embedding $\mathbf{w}_i^w \in \mathbb{R}^{d^w}$ by the BERT model (light blue vectors in Figure 5.1A). d^p -dimensional position embeddings \mathbf{w}_i^{p1} and \mathbf{w}_i^{p2} for each wordpiece, which correspond to the relative positions from the first and second target mentions are prepared, respectively. (green vectors in Figure 5.1A) The wordpiece embedding \mathbf{w}_i^w and the position embeddings \mathbf{w}_i^{p1} and \mathbf{w}_i^{p2} are concatenated as in the following Equation (5.1):

$$\mathbf{w}_i = [\mathbf{w}_i^w; \mathbf{w}_i^{p1}; \mathbf{w}_i^{p2}], \quad (5.1)$$

where $[;]$ denotes concatenation. the resulting embeddings are used to prepare the input to the convolution layer.

\mathbf{z}_i that is the concatenation of k input embeddings² around w_i is introduced:

$$\mathbf{z}_i = [\mathbf{w}_{[i-(k-1)/2]}^T; \dots; \mathbf{w}_{[i-(k+1)/2]}^T]^T. \quad (5.2)$$

Convolution to the embeddings is applied as follows:

$$m_{i,j} = f(\mathbf{W}_j^{sent} \odot \mathbf{z}_i + b^{sent}), \quad (5.3)$$

where \odot is an element-wise product, b^{sent} is a bias term, and $f(\cdot)$ is a GELU [56] function.³

A weight tensor for convolution is defined as $\mathbf{W}^{sent} \in \mathbb{R}^{d^c \times (d^w + 2d^p) \times k}$. The j -th column of \mathbf{W}^{sent} is represented as \mathbf{W}_j^{sent} . k is a window size. The tensor \mathbf{W}^{sent} is depicted as a red box in the left part of Figure 5.1A. Then, max-pooling to convert the output of each filter in the convolution layer into a fixed-size vector is employed as follows:

$$\mathbf{h}^{sent} = \max_i m_{i,j}. \quad (5.4)$$

5.2.2 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. The wordpiece embeddings by BERT are directly used without word position embeddings to prepare the input to the convolution layer. A convolution weight tensor $\mathbf{W}^{desc} \in \mathbb{R}^{d^c \times (d^w) \times k}$ and bias b^{desc} for description are defined. Convolution and max-pooling are employed in the same way as the processing of the input sentences and the description representations \mathbf{h}^{desc1} and \mathbf{h}^{desc2} of drug mentions m_1 and m_2 are obtained respectively.

5.2.3 Molecular Structure Representation

The molecular graph structures of drugs are represented using GNNs. GNNs convert a drug molecule graph G into a fixed size vector h^g . Atoms as represented as nodes and

²Multiple windows can be employed instead of a single window with the size k , but there is no significant difference in the performance in the preliminary experiment.

³The GELU activation function is chosen from ReLU, eLU, SeLU and GELU based on the results in the preliminary experiment.

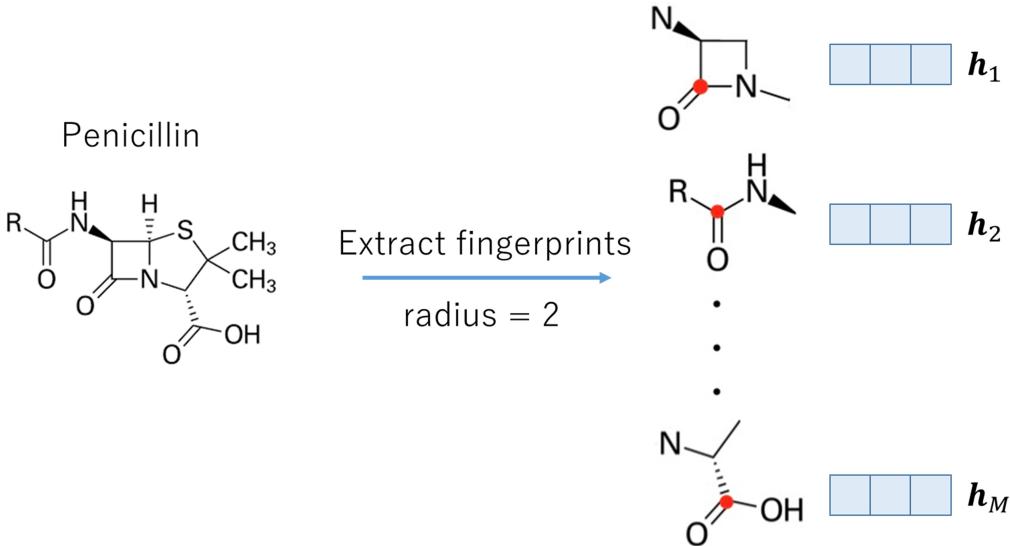


Figure 5.2: Illustration of molecular fingerprints. This figure shows the extraction of several fingerprint subgraphs from a molecular structure when radius is 2.

bonds as edges in the graph. The neural molecular GNN method proposed by Tsubaki et al. [53] is employed. 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. The vector of the i -th atom in a drug molecule is defined as \mathbf{h}_i and the set of its neighboring atoms as N_i . The vector \mathbf{h}_i is updated in the ℓ -th step as follows:

$$\mathbf{h}_i^\ell = \mathbf{h}_i^{\ell-1} + \sum_{j \in N_i} f(\mathbf{W}_{hidden}^{\ell-1} \mathbf{h}_j^{\ell-1} + \mathbf{b}_{hidden}^{\ell-1}), \quad (5.5)$$

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.

$$\mathbf{h}^{mol} = f(\mathbf{W}_{output} \sum_i^M \mathbf{h}_i^L + \mathbf{b}_{output}), \quad (5.6)$$

where M is the number of fingerprints. Figure 5.2 shows how the molecular GNN model extracts fingerprints including β -lactam (\mathbf{h}_1) from penicillin drug ($r=2$) and update fingerprint vectors.

The molecular structure representations \mathbf{h}^{mol1} and \mathbf{h}^{mol2} of drug mentions m_1 and m_2 are obtained, respectively.

5.2.4 DDI Extraction Using Database Information

When the drug description information for DDI extraction is used, the input sentence representation and two description representations as in Equation 5.7:

$$\mathbf{h} = [\mathbf{h}^{sent}; \mathbf{h}^{desc1}; \mathbf{h}^{desc2}]. \quad (5.7)$$

Similarly, two molecular structure representations are concatenated with the input sentence representation as in Equation 5.8:

$$\mathbf{h} = [\mathbf{h}^{sent}; \mathbf{h}^{mol1}; \mathbf{h}^{mol2}]. \quad (5.8)$$

The resulting vector is used as the input to the prediction layer. \mathbf{h} into prediction scores is converted using a weight matrix $\mathbf{W}^{pred} \in \mathbb{R}^{o \times d_p}$:

$$\mathbf{s} = \mathbf{W}^{pred}\mathbf{h}, \quad (5.9)$$

where $\mathbf{s} = [s_1, \dots, s_o]$ and o is the number of DDI types. \mathbf{s} is converted into the probability of possible interactions \mathbf{p} by a softmax function:

$$\mathbf{p} = [p_1, \dots, p_o], \quad p_j = \frac{\exp(s_j)}{\sum_{l=1}^o \exp(s_l)}. \quad (5.10)$$

The DDI extraction using drug description information and drug molecular structure information in Figure 5.1B and C are illustrated, respectively.

5.2.5 Training

The loss function L is defined as in the Equation (5.11) using \mathbf{p} in Equation (5.10) when the gold type distribution \mathbf{y} is given. \mathbf{y} is a one-hot vector where the probability of the gold label is 1 and the other probabilities are 0.

$$L = - \sum \mathbf{y} \log \mathbf{p} \quad (5.11)$$

5.2.6 Ensemble

An ensemble technique is employed to combine the prediction from different models. Specifically, the prediction scores are simply summed up from different models for the

ensemble after each of the models is trained separately. For instance, when the prediction of the model with the description information and that with the molecular structure information are combined, the prediction scores are summed up in Equation (5.9) as follows:

$$\mathbf{s} = \mathbf{s}^{desc} + \mathbf{s}^{mol}. \quad (5.12)$$

5.3 Experimental Settings

This section explains the DDI extraction task settings, drug database preprocessing, drug mention linking, and hyper-parameter settings.

5.3.1 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. The first sentence of the drug description of *Salbutamo* is shown 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.

To obtain the graph of a drug molecule, the SMILES string encoding of the molecule is obtained as input from DrugBank and then converted it into the graph structure using RDKit [57]. Fingerprints are extracted from the graph using preprocessing scripts provided by Tsubaki et al. [53].

5.3.2 Drug Mention Linking

Mentions in the corpus are linked to DrugBank entries by relaxed string matching. In particular, each mention and the following items in the DrugBank entries are lowercased, and the entry that includes an item showing the most overlap with the mention is chosen.

- Name: Headword of the drug entry
- Brand: Brand names from different manufactures

Corpus

*... a comprehensive review was made of **norgestrel** and **norethindrone**, ...*

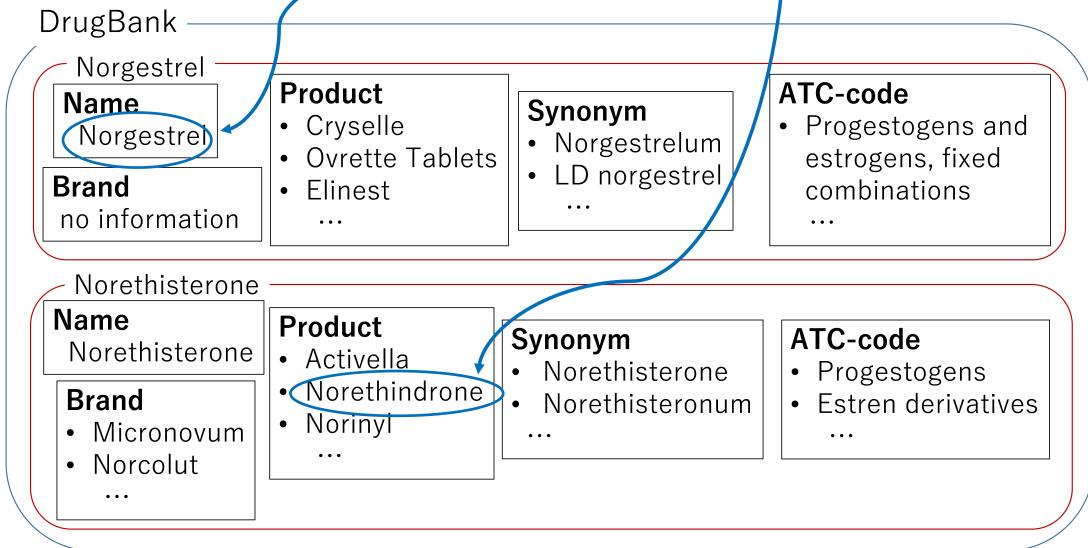


Figure 5.3: Linking between mentions and DrugBank entry

- 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 only the ATC codes that are assigned to single drugs is used for mention linking. Also, for synonyms, mentions and synonyms are linked 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 DDIExtraction-2013 train and test data set matched the DrugBank entries. Figure 5.3 shows how the linking is performed. The input sentence contains two mentions “norgestrel” and “norethindrone”. String matching is conducted to link these mentions to DrugBank entries. As a result, the mention “norgestrel” matched the Name item and the mention “norethindrone” matched the Product item.

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 5.1: Hyper-parameters for CNNs

5.3.3 Training Settings

This section follows the training settings for the fine-tuning of BERT on the GLUE tasks [4] except for the following two points. First, the AdamW optimizer [58] is employed instead of Adam optimizer. Second, mixed-precision training [59] is employed for the memory efficiency.

Dropout [60] is employed 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} . The description and molecular structure vectors of unmatched entities are set to zero vectors. Table 5.1 and 5.2 shows hyper-parameters for CNNs and GNNs. The same hyper-parameters are used as the GLUE tasks in Devlin et al. [4] for the BERT layer. In the DDIExtraction 2013 shared task, the official development data set is not provided; thus a development data set is prepared from the official training data set to choose the other hyper-parameters. In order to train the model on the same setting as other existing models [26, 43, 61], the development data set is included in the entire training data set for training the model. The entire training data set for training the model is used to evaluate the performance on the test set. For GNNs, the results with different radii 0, 1, and 2 for molecular fingerprints is shown. Note that GNNs with a radius of 0 mean no molecular fingerprints, which assign vectors to atoms.

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

Table 5.2: Hyper-parameters for GNNs

5.4 Results and Discussions

Table 5.3 shows the performance of DDI extraction models including the proposed models with different settings and the state-of-the-art models. It can be seen 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 it is compared with the state-of-the-art models in the top rows of the table. When the CNNs are omitted from the baseline model (SciBERT Linear), the first special token [CLS] is used as the aggregated representation of the sentence and 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.

Additional increase of the micro F-score is observed 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. The author leaves 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). The scikit-learn [62] Python library is used for evaluating the statistical significance.

Table 5.4 shows the performance of DDI extraction models on the development data set. Consistently with the results on the test set in Table 5.3, either of the description information and molecular structure information improves the performance and the combination of the two kinds of information showed the highest F-scores on the development data set. However, there are some inconsistencies in the results on development and test data sets; the model with molecular structure information showed a higher F-score than the model with description information on the development data set, while the model with molecular structure information showed a lower F-score on the test data set.

Table 5.5 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 kind of 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. The training data set is cross-validated using 5-fold cross-validation and further analyzed the performance on individual DDI types. Table 5.6 shows the F-scores for folds of cross-validated training data set. The micro-averaged F-score is used to calculate the average of the folds. The models with individual information source 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 source on average. As a result, the model

Method	P	R	F (%)
Liu et al. [26]	75.29	60.37	67.01
BioBERT [61]	-	-	78.8
Text-only (word2vec CNN)			
[43]	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

Table 5.3: Evaluation on DDI extraction from texts on the test set. Text only (SciBERT CNN) model is defined as the baseline model.

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 5.4: Evaluation on DDI extraction from texts on the development set

with the ensemble technique improves the F-scores on average for all the types except for *Int.*, where the model performs on par with the baseline model. These results show that the 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, the model is effective for improving the performance of DDI extraction.

Table 5.7 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.

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

Table 5.5: Performance on individual DDI types in F-scores. The best score for each type is shown in bold and the scores lower than the baseline model are shown with underlines.

5.4.1 Pre-training of GNNs and CNNs on DrugBank

To investigate the further use of DrugBank information, it is verified if the DrugBank DDI labels can improve the DDI extraction performance. Specifically, GNNs are pretrained 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.

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

First, the performances of the accuracy of binary classification on DrugBank DDI pairs are shown in Table 5.8. 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

			DDI Type		
	Method	Mech.	Effect	Adv.	Int. (%)
Fold 1	Text-only	84.60	86.38	85.80	68.29
	+ Desc	82.55	<u>81.82</u>	<u>85.23</u>	<u>64.37</u>
	+ Mol (radius=1)	<u>84.55</u>	<u>84.62</u>	<u>84.53</u>	71.05
	+ Desc + Mol (radius=1)	86.13	<u>85.46</u>	86.69	<u>67.47</u>
Fold 2	Text-only	83.46	83.26	78.80	81.48
	+ Desc	84.15	<u>82.52</u>	81.99	<u>79.01</u>
	+ Mol (radius=1)	<u>82.26</u>	83.45	81.64	<u>76.54</u>
	+ Desc + Mol (radius=1)	84.29	83.38	82.64	<u>79.01</u>
Fold 3	Text-only	84.91	59.21	76.54	91.43
	+ Desc	<u>83.40</u>	88.31	<u>73.53</u>	91.67
	+ Mol (radius=1)	<u>84.43</u>	86.24	<u>75.24</u>	94.44
	+ Desc + Mol (radius=1)	86.09	87.25	<u>76.22</u>	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	<u>77.34</u>	<u>76.92</u>
	+ 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	<u>87.73</u>	81.48
	+ Mol (radius=1)	86.09	83.74	<u>87.23</u>	<u>73.33</u>
	+ Desc + Mol (radius=1)	86.26	84.91	<u>88.34</u>	<u>75.00</u>
Average	Text-only	82.34	76.99	81.67	79.07
	+ Desc	83.09	84.39	<u>81.27</u>	<u>78.09</u>
	+ Mol (radius=1)	82.47	83.57	<u>81.60</u>	<u>78.97</u>
	+ Desc + Mol (radius=1)	84.01	85.20	82.70	<u>78.99</u>

Table 5.6: Individual F-scores on 5-fold cross-validated training data set. The micro-averaged F-score is used 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.

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.

CNNs and GNNs are pre-trained using the DrugBank interaction labels including the pseudo negative labels and fine-tuned on the DDIExtraction 2013 data set. Table 5.9 shows the comparison of the F-scores with or without pre-training. 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.

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 5.7: Comparisons of F-scores on different parts of the test set

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

Table 5.8: Accuracy of binary classification on the DrugBank pairs

5.4.2 Can DrugBank Information Alone Extract DDIs from Texts?

To further investigate how contextual information is important in the DDI task, it is verified whether the textual DDI can be extracted only from the drug information in DrugBank without using the input sentence. The input sentence representation \mathbf{h}^{sent} is simply omitted from Equation 5.7 and 5.8 and the DDI extraction models are trained, but the F-scores were quite low ($\sim 5\%$) for both models. This result shows that DDI relations cannot be extracted 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 the models on the database information serve as a supplement to the textual CNN model.

5.4.3 Error Analysis

Figure 5.4 shows F-scores for different sentence lengths on the validation data set. Since the instances with longer sentence lengths are relatively few, 5-fold cross-validation is used on the official training data set. Here, the sentence length is defined to be the number of subwords divided by the SciBERT vocabulary. In the previous work, Quan et al. [41] 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. [63] also analyzed the F-scores

	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

Table 5.9: Evaluation on DDI extraction from texts with or without pre-training of GNNs for the molecular structure and CNNs for the description

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. The model shows higher performance than the baseline model, especially for 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.

5.5 BioCreativeVII Track-1 DrugProt

5.5.1 Introduction

The DrugProt task of the BioCreative VII Track 1 is tackled with neural models that employ external knowledge. The models are based on BERT, which shows the state-of-the-art performance on several NLP tasks and can be considered as external knowledge from other texts. In addition, distant supervision data [64] is utilized and information of structure of drugs and proteins is utilized as external knowledge from knowledge bases.

5.5.2 Task Definition

The DrugProt data set is a corpus that is exhaustively annotated by domain experts, and all drug and protein mentions in PubMed articles are labeled. In addition, for all possible drug-protein pairs, binary relationships corresponding to the 13 types of drug-protein interactions are annotated. In other words, when some binary relations are true,

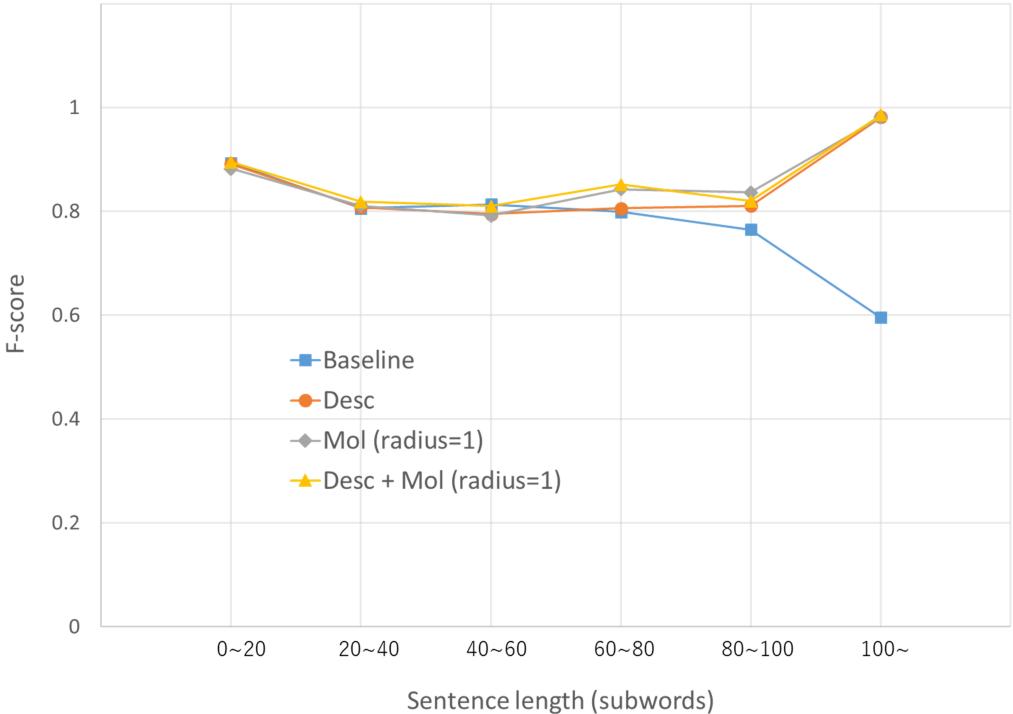


Figure 5.4: F-scores for different sentence lengths on the 5-fold cross-validated training data set. The micro-averaged F-score is used to calculate the average of the folds.

the drug-protein pair has multiple interactions and when all binary relations are false, it indicates that there is no interaction between the pair. The goal of the task is to correctly predict the interaction between drug-protein pairs given the input sentence and the mentions of drug and protein.

5.5.3 Methods

The model that utilizes the description and structure of the protein and drug entities is proposed.

Utilizing descriptions and structures of entities The first model utilizes the descriptions and structures of the protein and drug entities. This model is based on the drug-drug interaction extraction method of Asada et al. [9] and the model for drug-protein interaction extraction has been extended. The drug and protein mentions in an input sentence are linked to the databases DrugBank [39] and Uniprot [65], respectively. The textual information and information of structure registered in the database are then used for

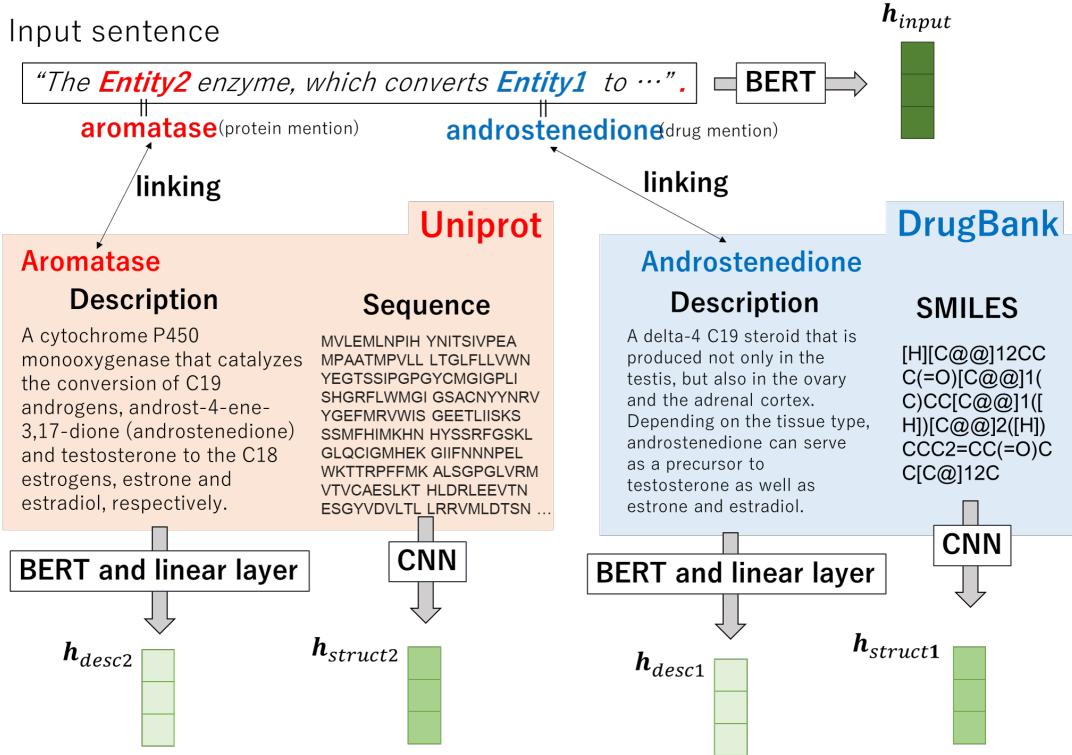


Figure 5.5: The model with description and structure information

relation extraction. Figure 5.5 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” [48] item is used as information of structures. For each protein, the “function” item registered in Uniprot is used as the description information and the “sequence” item is used as the information of structures.

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 h_{input} . h_{input} is then taken as the input of the fully connected layer and obtains the d_r -dimensional vector h_{input}^{fc} , where d_r is the number of relation labels including the negative label. It should be noted here that although DrugProt data set contains multiple labels for an instance, the approach cannot predict the multiple labels correctly. However, since it is known that the number of instances of multiple labels is extremely small in the exploratory experiment, such a standard classification approach is conducted.

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

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. The vector \mathbf{h}_{desc_CLS} of the BERT [CLS] token is converted into a d_d -dimensional vector \mathbf{h}_{desc} as follows:

$$\mathbf{h}_{desc} = \text{GELU}(\mathbf{W}\mathbf{h}_{desc_CLS} + \mathbf{b}), \quad (5.13)$$

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

$$\mathbf{h}_{desc}^{fc} = \text{FC}([\mathbf{h}_{input}; \mathbf{h}_{desc1}; \mathbf{h}_{desc2}]), \quad (5.14)$$

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

Encoding information of structures For drugs, the SMILES strings are used as the information of structures. For proteins, the amino acid sequences are used. Both SMILES strings and amino acid sequences are encoded by character-based CNNs.

First, the d_c -dimensional character embedding is assigned to each character of the 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 [66], and max pooling is employed to obtain the whole sequence representation. The representation of the entity 1 structure representation $\mathbf{h}_{struct1}$, the entity 2 structure representation $\mathbf{h}_{struct2}$, and the input sentence representation \mathbf{h}_{input} are concatenated to make the input of the fully connected layer:

$$\mathbf{h}_{struct}^{fc} = \text{FC}([\mathbf{h}_{input}; \mathbf{h}_{struct1}; \mathbf{h}_{struct2}]), \quad (5.15)$$

\mathbf{h}_{struct}^{fc} is converted into the probability \mathbf{p}_{struct} by a softmax function, and update the model parameters by minimizing the cross-entropy loss.

Inference Finally, description and structure information are combined 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:

$$\mathbf{p}_{all} = \frac{1}{3}(\mathbf{p}_{input} + \mathbf{p}_{desc} + \mathbf{p}_{struct}). \quad (5.16)$$

The relation label prediction is calculated as $\text{argmax}\mathbf{p}_{all}$.

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 5.10: Micro-averaged F-scores on DrugProt 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.

5.5.4 Experiments

Models The five models have been submitted for BioCreative VII competition:

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

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-A.	0.000	0.000	0.000	0.571	0.000
AGONIST-I.	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-D.	0.758	0.772	0.742	0.747	0.778
INDIRECT-U.	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_P.	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

Table 5.11: F-scores per class on DrugProt development set. AGONIST-A., AGONIST-I., INDIRECT-D., INDIRECT-U. and SUBSTRATE_P. stand for AGONIST-ACTIVATOR, AGONIST-INHIBITOR, INDIRECT-DOWNREGULATOR, INDIRECT-UPREGULATOR and SUBSTRATE_PRODUCT-OF, respectively.

2-ds_desc_struct The ensemble of model 1 and 4

3-ds_init_desc_struct The ensemble of model 1 and 5

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

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

Experimental settings For the model **1-desc_struct**, BioBERT-Large [67] is used as the text encoder. 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. The dimension size d_d was set to 32.

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 train data set matched the DrugBank and Uniprot entries. When the entity could not be linked to database entries or the description and structure information is not registered in the database, an empty string is used as the input of BERT and CNNs, that is, all tokens are replaced with the padding token.

In the information of structures 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 three convolution windows are used, the dimension sizes of the structure vectors $\mathbf{h}_{struct1}$ and $\mathbf{h}_{struct2}$ are both $16 \times 3 = 48$.

Results The performance of proposed models is evaluated on the development set and test set in Table 5.10. 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 cases where the description information and the structure information is used individually, compared with the baseline BioBERT-Large model. The model **1-desc_struct**, which uses both description and structure information, further improved the F-score from baseline model.

Table 5.11 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 with weights pre-trained on distantly supervised data, showed higher performance.

5.6 Summary

A novel neural method for relation extraction from text using large-scale raw text information and drug database information, especially the drug descriptions and the drug

molecular structure information, is proposed. The results show that the large-scale raw text information with SciBERT greatly improves the performance of DDI extraction from texts on the data set of the DDIExtraction-2013 shared task. In addition, either the drug descriptions and the molecular structures can further improve the performance for specific DDI types, and their simultaneous use can improve the performance on all the DDI types.

6 Representing Heterogeneous Knowledge Graph

This chapter includes work from the published paper Asada et al. (2021) [10].

6.1 Background: An Overview of Knowledge Graph Representation

Recently, obtaining the representation of Knowledge Graph (KG) elements in a dense vector space has attracted a lot of research attention. Major advances in the KG representation learning model, which expresses entities and relationships as elements of a continuous vector space, are witnessed. The vector space embedding of all elements in KGs has received considerable attention because it is used to create a statistical model of the whole KGs, i.e., to easily calculate the semantic distance between all elements and to predict the probabilities of possible relational events (i.e., edges) on the graph. Such models can be used to infer new knowledge from known facts (i.e., link prediction), to clarify entities (i.e., entity resolution), to classify triples (i.e., triple classification), and to answer the probability question and answering [68–71]. They can enhance knowledge learning capabilities from the perspectives of knowledge reasoning, knowledge fusion, and knowledge completion [72–75].

Applications of the KG are often severely affected by data sparseness; however, a typical large-scale KG is usually far from perfect. The task of completing the KG aims to enrich the KGs with new facts. Many graph-based methods have been proposed to find new facts between entities based on the network structure of KG [76]. Much effort has also been put into extracting relevant facts from plain text [21]. However, these approaches do not utilize KG information. Neural-based knowledge representations have been proposed to encode both entities and relationships in a low-dimensional space where new facts can be found in [77, 78]. While traditional methods often deal with KGs without node types, in many real world data, entities have different semantic types. Recent methods deal with heterogeneous KGs with different types of nodes [79, 80]. More importantly, neural

models can be used to perform learning of text and knowledge within a unified semantic space to more accurately complete the KG [81].

Nowadays, there has been a lot of interest in jointly learning KG and embedding textual information. However, traditional KG models based on representation learning only use the information of molecular structures embedded in a particular KG. Plain text textual information, on the other hand, provides a wealth of semantic and contextual information that can contribute to the clarity and completion of entity representations and relationship representations of a given KG. Therefore, textual information can be seen as an effective supplement to the completed task of the KG. To explore the informative semantic signals of plain text, there has recently been a great deal of interest in learning together the embeddings of KG and text information in [82]. Moreover, the researchers provided a text-enhanced KG representation model that utilized textual information to enhance the knowledge representations [83].

KGs have attracted great attention from both academia and industry as a means of representing structured human knowledge. Various kinds of KGs have been proposed such as Freebase [84], YAGO [85], and WordNet [86]. A KG is a structured representation of facts that consists of entities, relations, and semantic descriptions. Entities are real-world objects and abstract concepts, relations represent relationships between entities, and semantic descriptions of entities, and these relationships include types and properties that have well-defined meanings. The KG usually consists of a set of triples $\{(h, r, t)\}$, where h , r , and t represent the head entity, relationship, and tail entity, respectively.

Based on the above motivation, this chapter investigates a heterogeneous pharmaceutical knowledge graph containing textual information constructed from several databases. The heterogeneous entity items consisting of drug, protein, category, pathway, and Anatomical Therapeutic Chemical (ATC) code, and relations among them, which include category, ATC, pathway, interact, target, enzyme, carrier, and transporter, are constructed. Three methods are compared to incorporate text information in KG embedding training with

representing text with BERT. The resulting node and edge embeddings are evaluated by the link prediction task and the usefulness of using text information in KG embedding training is verified. The study of KG completion is roughly divided into two types: a study in which the link prediction task is performed by using score functions such as TransE [78], DistMult [87], and a study in which Graph Convolutional Networks [88] etc, are applied to the whole KG, and the node classification task is performed. In this study, the link prediction task is in focus and the usefulness of text information in scoring function-based link prediction tasks is investigated.

The contributions are summarized as follows:

- A heterogeneous KG with textual information (called *PharmaHKG*) in the drug domain is proposed. This can be used to develop and evaluate heterogeneous knowledge embedding methods.
- Three methods to incorporate text information into KG embedding models are proposed.
- The combinations of four KG embeddings models and three methods are evaluated and compared on the link prediction task in the proposed KG, and it is shown that there is no single method that can perform best for different relations and the best combination depends on the relation type.

6.2 Heterogeneous Pharmaceutical Knowledge Graph with Textual Information

In this section, a heterogeneous pharmaceutical KG PharmaHKG that is constructed in this paper is first introduced. Then, the definition of KG and the learning method of embeddings in the KG are explained. Finally, the proposed method that effectively uses text information for KG representation learning is explained.

A heterogeneous pharmaceutical KG with textual information is constructed from DrugBank [39] and its related data sources. DrugBank is one of the rich drug databases.

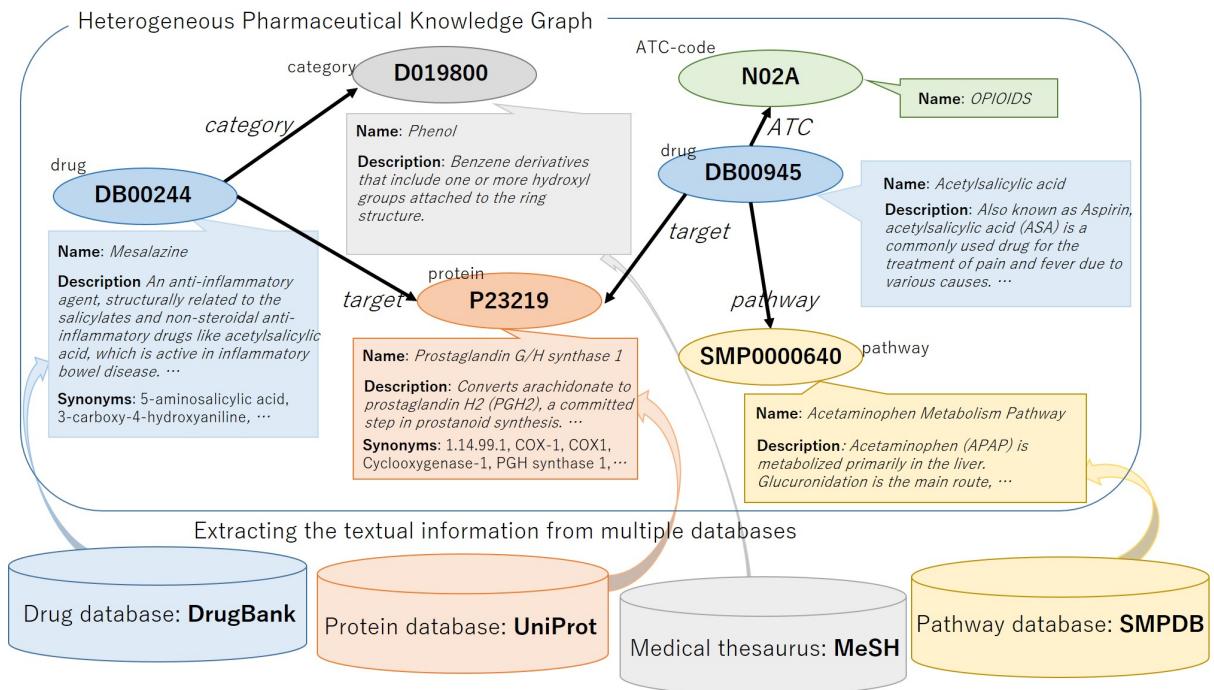


Figure 6.1: Illustration of the Heterogeneous Pharmaceutical Knowledge Graph

It contains several different types of nodes, which can be a good source for a heterogeneous knowledge graph. The nodes are related to several textual information in DrugBank and their linked entries in several other data sources such as UniProtKG [65], Small Molecule Pathway Database (SMPDB) [89] and medical thesaurus Medical Subject Headings (MeSH). The existence of such textual information fits the objective to evaluate the utility of textual information in KG representation. The KG and the related data sources are illustrated in Figure 6.1. This section first explains the nodes and relations in the KG and then explains the textual information.

6.2.1 Constructing Heterogeneous Pharmaceutical Knowledge Graph

A KG consisting of five different types of heterogeneous items, i.e., drug, protein, pathway, category, and ATC code, is constructed from different databases and thesaurus. The number of nodes is shown in Table 6.1.

- **Drug**: information of drugs is extracted from DrugBank [39]. More than 10,000 drugs are registered in DrugBank, and various types of information such as drug names, descriptions, molecular structures and experimental properties are registered.

Entity type	#
Drug (DrugBank-ID)	11,516
Protein (Uniprot-ID)	5,339
Pathway (SMPDB-ID)	874
Category (MeSH-ID)	2,166
ATC (ATC-code)	1,093
Total	20,988

Table 6.1: Statistics of heterogeneous pharmaceutical KG entities

- **Protein:** The information of proteins is extracted from UniProtKG [65]. UniProtKG consists of the Swiss-Prot which is manually annotated and reviewed and TrEMBL which is automatically annotated and not reviewed, and the Swiss-Prot knowledge base is used.
- **Pathway:** Information of pathways from Small Molecule Pathway Database (SMPDB) [89] is extracted. SMPDB is an interactive, visual database containing more than 30,000 small molecule pathways found in humans.
- **Category:** Information of drug categories is extracted from medical thesaurus Medical Subject Headings (MeSH) [90]. Each drug recorded in DrugBank has several hypernymy categorical classes and these classes have MeSH term ID. As an example, a drug *Morphine* has categories such as *Alkaloids* (MeSH ID:D000470), *Anesthetics* (MeSH ID:D018681), and the detailed information can be obtained by referring to MeSH.
- **ATC:** Anatomical Therapeutic Chemical (ATC) classification system also has categorical information of drugs. In the ATC classification system, drugs are divided into different groups according to the organ or system on which they act and their therapeutic, pharmacological, and chemical properties. Drugs are classified in groups at five different levels. The drugs are divided into fourteen main groups (first level), with pharmacological or therapeutic subgroups (second level). The third and fourth levels are chemical/pharmacological/therapeutic subgroups and the fifth level is the

Relation type	ALL	train	valid	test
category	60,459	54,419	3,020	3,020
ATC	16,341	14,711	815	815
pathway	18,707	16,847	930	930
interact	2,682,142	2,413,932	134,105	134,105
target	18,467	16,627	920	920
enzyme	5,206	4,686	260	260
carrier	815	735	40	40
transporter	3,093	2,793	150	150
Total	2,750,228	2,525,829	140,240	140,240

Table 6.2: Statistics of heterogeneous pharmaceutical KG edges for each relation type

chemical substance. For example, the drug “Metformin” is classified into “A: Alimentary tract and metabolism” (first level), “A10: Drugs used in diabetes” (second level), “A10B: Blood glucose lowering drugs, excl. insulins” (third level), “A10BA: Biguanides” (fourth level) and “A10BA02: metformin” (fifth level).

Five different types of nodes are connected by the following eight types of relations: *category*, *ATC*, *pathway*, *interact*, *target*, *enzyme*, *carrier*, and *transporter*. The statistics of the KG edges for each relation type are shown in Table 6.2. The relation triples from DrugBank are extracted.

Drug nodes and MeSH categorical terms are linked by *category* relation.

- *category*: This relation type indicates the MeSH category of drugs. This relationship indicates that the drug is classified into the therapeutic category or the general category (anti-convulsant, antibacterial, etc.) defined by MeSH. These relationships are registered by the manual search of DrugBank developers.

Drug nodes and ATC classification system codes are linked by *ATC* relation. In order to incorporate hierarchical information into the KG, ATC codes are linked to ATC codes by *ATChypernym* relation. ATC codes are linked to the next higher level codes with this relation. Relational triples such as A10BA - ATChypernym - A10B, N02 - ATChypernym - N by linking the ATC code of the next higher level are created. Since this relation is apparent from the surface strings of ATC codes, this relation for link prediction is not considered.

- *ATC*: Drugs are linked to any level of ATC codes with this relation. In DrugBank, drug elements may have one or more ATC-code elements, e.g., drug *Morphine* has four ATC codes (A07DA52, N02AA51, N02AA01 and N02AG01), and each ATC-code element has child elements. All these child entities and the drug entity are connected by the *ATC* relation.

Drug nodes and protein nodes are also connected with pathways.

- *pathway*: This relation type indicates a drug or protein is included in a pathway. When the drug is involved in metabolic, disease, and biological pathways as identified by the SMPDB, the drug entity and the pathway entity is connected by the *pathway* relation. Also, when the enzyme protein is involved in the same pathways, the protein entity and the pathway entity are connected by the *pathway* relation.

Drug nodes can be connected by a relation *interact*.

- *interact*: A triple of this relation type indicates that the drug pair has a DDI. When concomitant use of the pair of drugs will affect its activity or result in adverse effects, these two drug entities are connected by *interact* relation. These interactions may be synergistic or antagonistic depending on the physiological effects and mechanism of action of each drug.

Drug nodes and protein nodes can be linked by *target*, *enzyme*, *carrier*, or *transporter* relation [39].

- *target*: A protein, macromolecule, nucleic acid, or small molecule to which a given drug binds, resulting in an alteration of the normal function of the bound molecule and a desirable therapeutic effect. Drug targets are most commonly proteins such as enzymes, ion channels, and receptors.
- *enzyme*: A protein that catalyzes chemical reactions involving a given drug (substrate). Most drugs are metabolized by the Cytochrome P450 enzymes.

- *carrier*: A secreted protein that binds to drugs, carrying them to cell transporters, where they are moved into the cell. Drug carriers may be used in drug design to increase the effectiveness of drug delivery to the target sites of pharmacological actions.
- *transporter*: A membrane bound protein that shuttles ions, small molecules, or macromolecules across membranes, into cells or out of cells.

6.2.2 Textual Information of Knowledge Graph

Here, the text information relating to each type of node is explained.

- **Drug**: Drugs are assigned a unique DrugBank-id. Various text information contained in the DrugBank xml file is used. “Name”, to the heading of the drug and standard name of the drug as provided by the drug manufacturer, “Description”, which describes the general facts, composition and/or preparation of the drug, “Indication” is a description or common names of diseases that the drug is used to treat, “Pharmacodynamics” is a description of how the drug works at a clinical or physiological level, “Mechanism of Action” is a description of how the drug works or what it binds to at a molecular level, “Metabolism” is a mechanism by which or organ location where the drug is neutralized, and “Synonyms” indicates alternate drug names.
- **Protein**: Protein targets of drug actions, enzymes that are inhibited/induced or involved in metabolism, and carrier or transporter proteins involved in the movement of the drug across biological membranes. Each of *targets*, *enzymes*, *carriers*, *transporters* have unique UniProt-id. UniProt-id is referred to and the following types of textual information, functions, miscellaneous description, short name, alternative names, and gene names are obtained.
- **Pathway**: Pathway relations are extracted from DrugBank. Each pathway has

a unique ID defined by The Small Molecule Pathway Database (SMPDB) [89].

“Name” and “Description” of the pathway are registered in SMPDB.

- **Category:** Drug categories are classified according to the medical thesaurus MeSH.

This textual information is registered in MeSH: “Name” is a definition word, “ScopeNote” is a term description, “Entry terms” is a synonym.

- **ATC:** Drugs are classified in a hierarchy with five different levels by WHO drug classification system (ATC) identifiers. Each level of ATC classification code has a name, which is defined as the international nonproprietary name (INN) or the name of the ATC level. These names given to ATC codes as textual information are used.

6.3 Learning Knowledge Graph Embeddings

6.3.1 Knowledge Graph Definition

A heterogeneous KG is treated as a directed graph whose nodes and edges have semantic types. The semantic types are assigned to different types of nodes (drug, protein, pathway, etc.) and relations (target, carrier, etc.) to represent detailed information about nodes and relations. A KG is defined as a directed graph $\mathcal{G} = (E, R, F)$, where the nodes E denotes the set of typed entities, R refers to the set of typed relations and F represents the set of facts (i.e., directed edges). The nodes are often called entities. The facts or directed edges are often called triplets and are represented as a (h, r, t) tuple, when h is the head entity, t is the tail entity and r is the relation from the head entity to the tail entity.

6.3.2 Scoring Functions

The methods that represent the KG by using embeddings of entities and relations can catch the structure information of the KG and provide structure-based embeddings. Entities and relations are directly represented as real-valued vectors, matrices or complex-valued vectors. Scoring function $f(h, r, t)$ is defined on each triple (h, r, t) to access the validity of triples. Triples observed in the KG tend to have higher scores than those that

have not been observed. The following four scoring functions are employed.

TransE TransE [78] is a representative translational distance model that represents entities and relations as vectors in the same semantic space of dimension \mathbb{R}^d where d is the dimension of the target space with reduced dimension. A fact in the source space is represented as a triplet (h, r, t) . The relationship is interpreted as a translation vector so that the embedded entities are connected by relation r and have a short distance. The norm is set to 2, and the scoring function is computed as:

$$f(h, r, t) = -|\mathbf{h} + \mathbf{r} - \mathbf{t}|_2. \quad (6.1)$$

DistMult DistMult [87] is a method that speeds up the RESCAL model [68] by considering only symmetric relations and restricting M_r from a general asymmetric $r \times r$ matrix to a diagonal square matrix, thus reducing the number of parameters per relation to $O(d)$. DistMult scoring function is computed as:

$$f(h, r, t) = \mathbf{h}^\top \text{diag}(\mathbf{r}) \mathbf{t} = \sum_{i=0}^{d-1} [\mathbf{h}]_i [\mathbf{r}]_i [\mathbf{t}]_i. \quad (6.2)$$

ComplEx ComplEx [91] uses complex vector operations to consider both symmetric and asymmetric relation. The scoring function for complex entity and relation vectors \mathbf{h} , \mathbf{r} , and $\mathbf{t} \in \mathbb{C}^d$ is computed as:

$$f(h, r, t) = \text{Real}(\mathbf{h}^\top \text{diag}(\mathbf{r}) \mathbf{t}), \quad (6.3)$$

where Real extracts the real part of the complex vectors.

SimplE SimplE [92] considers two vectors $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ as the head and tail embeddings for each entity and two vectors $\mathbf{v}_r, \mathbf{v}_{r^{-1}} \in \mathbb{R}^d$ for each relation r . The similarity function of SimplE for a triple (h, r, t) is defined as:

$$f(h, r, t) = \frac{1}{2} (\langle \mathbf{h}_h, \mathbf{v}_r, \mathbf{t}_t \rangle + \langle \mathbf{h}_t, \mathbf{v}_{r^{-1}}, \mathbf{t}_h \rangle). \quad (6.4)$$

The above four score functions are chosen because these are widely used and cover the standard ideas for scoring relational triples: distance-based, bilinear-based and complex number-based.

6.3.3 Negative Sampling and Loss Functions

Generally, to train a KG embedding, the models apply a variety of negative sampling by corrupting triplets (h, r, t) . They corrupt either h , or t by sampling from the set of head or tail entities for heads and tails, respectively. The corrupted triples can be either of (h', r, t) or (h, r, t') , where h' and t' are the negative examples. The author acknowledges that due to the incompleteness of the current KG, the unregistered and potentially positive relational triples can be negative examples: this problem is common in most studies that tackle with the link prediction task. To avoid easy negative examples and utilize the entity type information, the node types of negative examples is restricted depending on r . The logistic loss and the margin-based pairwise ranking loss are commonly used for training. The logistic loss returns -1 for negative examples and $+1$ for the positive examples. \mathbb{D}^+ and \mathbb{D}^- are negative and positive data, $y = \pm 1$ is the label for positive and negative triples, and $f(\cdot)$ is the scoring function. Model parameters are trained by minimizing the negative log-likelihood of the logistic model with $L2$ regularization on the parameters Θ of the model;

$$L_{KG} = \sum_{(h, r, t) \in \mathbb{D}^+ \cup \mathbb{D}^-} \log(1 + \exp(y \times f(h, r, t))) + \lambda \|\Theta\|_2^2. \quad (6.5)$$

The margin-based pairwise ranking loss minimizes the rank for positive triples. Ranking loss is given by:

$$L_{KG} = \sum_{(h, r, t) \in \mathbb{D}^+} \sum_{(h', r', t') \in \mathbb{D}^-} \max(0, \gamma - f(h, r, t) + f(h', r', t')) + \lambda \|\Theta\|_2^2. \quad (6.6)$$

6.4 Methods

This section verifies the usefulness of using text information in KG embedding training by three methods explained below. Figure 6.2 shows the overview of the three meth-

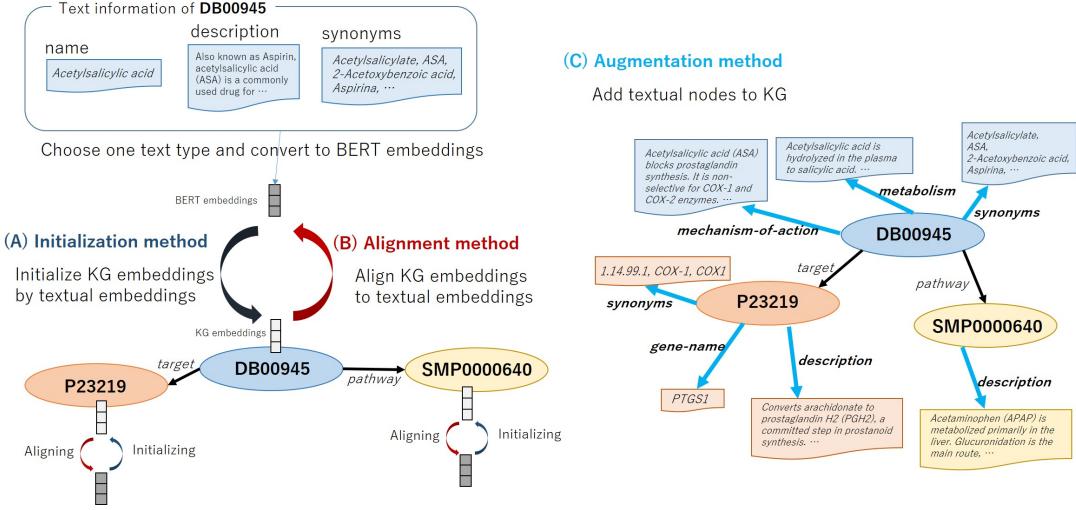


Figure 6.2: Overview of methods: (A) Initializing node embeddings (Initialization), (B) Aligning entity embeddings and textual embeddings (Alignment), and (C) Augmenting KG embeddings (Augmentation)

ods that utilize text information for KG embedding representation. BERT [4], which is an extremely high-performance contextual language representation model, is employed in encoding text. BERT is pre-trained with the masked language model objective and next sentence prediction task objective on large unlabeled corpora, and fine-tuned BERT towards the target task achieved the state-of-the-art performance.

6.4.1 Initializing Node Embeddings

Usually, the initial value of embedding for each node in the KG is given randomly in the existing methods. As shown in Figure 6.2 (A), first, the type of text to use, e.g., drug nodes have text types such as Name, Description and Synonyms, is selected. Then, the selected text is taken as the input of the text encoder model BERT and the <CLS> embeddings of the BERT as the initial value of the node embeddings. For the methods that use two embeddings for an entity, i.e., ComplEx (real and imaginary embeddings) and SimpleE (head and tail embeddings), both vectors with the <CLS> embeddings are initialized. When multiple text items are registered (e.g., the drug Acetaminophen has multiple synonyms, “Acenol”, “APAP”, “Paracetamol”), these terms are connected with a comma and taken as an input for BERT. This method is called the *Initialization* method.

The motivation of the *Initialization* method is to help represent node embeddings by using the BERT embeddings that pre-trained on a large amount of biomedical literature. Correct relational triples are aimed to predict from textual information by BERT even if the information of structures in the graph is insufficient.

6.4.2 Aligning Entity Embeddings and Textual Embeddings

The aligning method aims to gradually project KG embeddings into textual embeddings space by adding the regularization term to the loss function.

$$L_a = \lambda_a \|V_{KG} - V_{text}\|_2, \quad (6.7)$$

$$L = L_{KG} + L_a, \quad (6.8)$$

where λ_a is a regularization coefficient of alignment, V_{KG} and V_{text} are vector lookup table matrices of KG and textual embeddings respectively. Similar to the initialization method, the textual embeddings are obtained from BERT and when there are two embeddings for an entity, both vectors are regularized. This method is called the *Alignment* method. The motivation of the *Alignment* method is that as the updating the node representation progresses, the two spaces of the text embeddings and the graph structure embeddings are projected into the same space, and finally more suitable node representations are obtained.

6.4.3 Augmenting KG Embeddings

In this method, as shown in Figure 6.2 (C), the KG structure is augmented by adding relation triples based on the text information of the node. The node's own embedding is initialized with textual embeddings of Name. The embedding value of linked nodes is initialized with the BERT output. Moreover, since ATC classification codes have a hierarchical structure as shown in Figure 6.2 (C), after extending the link from the drug node to create new categorical nodes, further linking is made between the categorical nodes. A graph that can consider both text information and the hierarchical information

is constructed. This method is called the *Augmentation* method. The motivation of the *Augmentation* method is to consider multiple text information of one entity at once.

6.5 Experimental Settings

6.5.1 Constructing Heterogeneous KG with Textual Information

The overview of constructing a heterogeneous KG with textual information is shown in Figure 6.1. Four publicly available databases, DrugBank, UniProt, MeSH term descriptions and SMPDB are downloaded, and first DrugBank is processed and relations between the drug and other heterogeneous items are extracted. Here, the text information on each drug is also extracted and associated with the entity ID in the KG. Next, for entities other than drugs, the link ID of DrugBank is used to refer to other databases and associated the text information with the entity ID in the KG. As a result, five types of entities (i.e., drug, protein, pathway, category, and ATC) are included in the constructed KG. Between entities, there are relation links: category, ATC, pathway, interact, target, enzyme, carrier, and transporter. The total number of relational triples is about 2.7M, and as shown in Table 6.2, the number of drug-interact-drug triples is large and accounts for the majority of them. Note that only the relation drug-interact-drug is symmetric, and the other relations are asymmetric, that is, when there is a DrugA-interact-DrugB relation triple in the KG, there is also a DrugB-interact-DrugA triple.

6.5.2 Encoding Text Information

PubMedBERT [93] is employed to encode textual information into fixed-length real-valued embeddings. PubMedBERT is a model that uses 21B words of the PubMed corpus for pre-training, and shows high performance in several NLP tasks in the biomedical domain. In this paper, texts such as names and descriptions are used as inputs for pre-trained PubMedBERT, and the output <CLS> token embedding is used as a textual representation. The maximum length of the input subword is set to 512.

	Name (%)	Description (%)	Synonyms (%)
drug	100	53.72	48.50
protein	100	96.17	100
category	100	94.01	91.42
ATC	100	-	-
pathway	100	100	-

Table 6.3: The percentage of nodes that have each type of text. Nodes in all databases have Name text information. While many proteins and categories have Description information and Synonyms information, the percentage of drugs that have this information is low.

6.5.3 KG Embedding Training Settings

Four KG embedding scoring functions are employed as explained in Section 6.3.2. For each of the scoring functions, three methods are applied to train embeddings using textual information; the initialization, aligning and augmenting methods.

The ratio of nodes that have each textual information is shown in Table 6.3. The node has the text information of Name in any database. In UniProt, most proteins have Description and Synonyms texts information, and many categorical terms in MeSH also have Description and Synonyms. On the other hand, some drugs in DrugBank do not have some text information. In the Initialization method and Alignment method, one text type is selected and the embeddings of textual information are used⁴. When the node does not have text information, the text of Name is used instead.

Drugs and proteins have textual information that other nodes do not have, and their coverage is as follows: 32.61% of drugs have Indication information, 24.60% of drugs have Pharmacodynamics information, 18.40% of drugs have Metabolism information, 30.52% of drugs have Mechanism-of-action information and 96.05% of proteins have Gene-name. These text items are linked to the KG nodes in the Augmentation method, so the Augmentation method can utilize all text information.

The random initialization method is prepared without textual information (*No Text*) as the baseline. In this setting, embeddings of entities and relations are initialized with the random values drawn from a uniform distribution between $\pm(\gamma + \frac{\epsilon}{d})$, where $\gamma = 12$,

⁴the combination of different text information in these methods is left for future work

$\epsilon = 2$ and d is a dimension of KG embeddings.

6.5.4 Task Setting

The quality of node and edge embeddings are evaluated by the link prediction task. Link prediction is a task to search for an entity that probably constructs a new fact with another given entity and a specific relation. For KGs are always imperfect, link prediction aims to discover and add missing knowledge into it. With the existing relations and entity, candidate entities are selected to form a new fact. The head or tail of the triples in the validation or test data set are replaced with other entities that have the same entity types and calculate the scores of all created negative triples in the KG. The calculated positive triple score and the scores of all negative triples are sorted and the rank of the positive triple score is evaluated. Mean reciprocal rank (MRR) is used as an evaluation metric. When negative example triples are created, if there are correct triples that exist in the KG, such triples from the ranking are excluded. This evaluation setting has been adopted in many existing studies as a **filtered** setting [78, 91, 92]. In addition, similar to the negative sampling setting during training, given the relational edge label, the node types of head or tail are trivial, so triples with inappropriate combinations of edge and node types are also excluded.

The extracted approximately 2.7M relational triples are divided into 90:5:5 as the train, valid and test data sets. In the augmentation method, relational triples created from textual nodes are added to the train data set.

6.5.5 Hyper-parameter Settings

Hyper-parameters are tuned by evaluating the MRR score on the validation set for each model. Hyper-parameters are chosen with following values: regularization coefficient $\lambda \in \{10^{-3}, 10^{-6}, 10^{-9}, 10^{-12}, 0\}$, alignment regularization coefficient $\lambda_a \in \{10^{-3}, 10^{-6}, 10^{-9}, 10^{-12}\}$, initial learning rate $\alpha_0 \in \{0.5, 0.25, 0.1, 0.05, 0.025, 0.01\}$, For the loss function, the pairwise hinge loss function is adopted for TransE and DistMult and the logistic loss function

for ComplEx and SimplE according to the setting of the original papers. The KG embedding dimension is set to 768 in order to match the dimension of the output of BERT embedding. For all models, the batch size is set to 4096 and the number of epochs to 100.

6.5.6 Implementation Details

All the models are implemented by using the PyTorch library [94], the DGL-KE library [95] for KG embeddings, and the transformers library [96] for BERT. The original DGL-KE implementation is modified in the following point. While DGL-KE samples negative examples from all combinations of entity pairs, the proposed model excludes impossible negative instances by restricting the types of entities by the relations (e.g., a drug-interact-category triple is not created for negative examples) as explained in Section 6.3.3.

6.6 Results and Discussions

Table 6.4 shows the comparison of link prediction MRR for each relation edge type, the macro-averaged MRR. While a micro-average MRR is calculated by directly calculating the MRR for all instances in the KG without considering the types, a macro-averaged MRR is calculated by first calculating the MRR for each type and then taking the average of the MRR scores. Since the constructed triples are highly imbalanced and the proportion of interact triples is large, models with high prediction performance of relation *interact* can result in high micro-averaged MRR. The macro-averaged MRR is reported to avoid the effect of this imbalance. For each scoring function, the comparison of performance between the models with and without text information is shown.

When the TransE algorithm is used, in the *category* types, the textual models improved MRR but in other relation triple types, the MRR decreased and the averaged MRR also decreased. Of the three methods that used text, the Initialization by synonyms embeddings method showed the highest macro-averaged MRR.

When the DistMult scoring function is used, the MRR decreased in *interact* and

TransE								
	No Text	Initialization			Alignment			Augmentation
		Name	Desc.	Syn.	Name	Desc.	Syn.	
category	0.1978	0.2117	0.2120	0.2231	0.2136	0.2059	0.1913	0.2239
ATC	0.2929	0.2695	0.2571	0.2495	0.3000	0.2973	0.2872	0.2571
pathway	0.6741	0.5674	0.5792	0.5793	0.6694	0.6711	0.6713	0.5473
interact	0.3109	0.2867	0.2845	0.2843	0.3103	0.3106	0.3108	0.2644
target	0.0802	0.0748	0.0811	0.0808	0.0808	0.0780	0.0821	0.0889
enzyme	0.3262	0.3067	0.3314	0.3090	0.3474	0.3564	0.3590	0.2926
carrier	0.4155	0.3078	0.2843	0.3679	0.4037	0.4044	0.4010	0.3459
transporter	0.3576	0.3194	0.3104	0.3182	0.3444	0.3308	0.3391	0.2866
Avg. (Macro)	0.3319	0.2930	0.2950	0.3015	0.3337	0.3318	0.3302	0.2883
DistMult								
	No Text	Initialization			Alignment			Augmentation
		Name	Desc.	Syn.	Name	Desc.	Syn.	
category	0.2539	0.2876	0.2797	0.2753	0.2679	0.2661	0.2586	0.2649
ATC	0.2428	0.2674	0.2899	0.2639	0.2612	0.2617	0.2698	0.2904
pathway	0.6792	0.5424	0.5542	0.6002	0.6524	0.6711	0.6615	0.4997
interact	0.7730	0.6338	0.5895	0.6302	0.7911	0.7868	0.7990	0.6113
target	0.0738	0.0866	0.0864	0.0947	0.0778	0.0758	0.0734	0.1049
enzyme	0.2501	0.2516	0.2358	0.2847	0.2247	0.2334	0.2140	0.2183
carrier	0.2023	0.2254	0.1640	0.1622	0.1369	0.1311	0.1649	0.2134
transporter	0.2293	0.2703	0.1840	0.2190	0.1969	0.1939	0.1708	0.2062
Avg. (Macro)	0.3380	0.3206	0.2979	0.3162	0.3261	0.3274	0.3265	0.3011
ComplEx								
	No Text	Initialization			Alignment			Augmentation
		Name	Desc.	Syn.	Name	Desc.	Syn.	
category	0.0905	0.3455	0.3495	0.3386	0.3302	0.0577	0.0611	0.3420
ATC	0.3326	0.3463	0.3623	0.3485	0.3271	0.3425	0.3407	0.3652
pathway	0.6956	0.6856	0.7051	0.7157	0.7220	0.6963	0.7323	0.6820
interact	0.8678	0.7632	0.7166	0.7802	0.8578	0.8189	0.8497	0.8230
target	0.0496	0.1116	0.1093	0.1153	0.0859	0.0640	0.0740	0.1243
enzyme	0.2103	0.2256	0.2512	0.2538	0.2245	0.1907	0.2097	0.2073
carrier	0.1533	0.1557	0.1817	0.1423	0.0994	0.1750	0.1462	0.1934
transporter	0.1942	0.3119	0.2667	0.2593	0.2151	0.2076	0.2362	0.2801
Avg. (Macro)	0.3242	0.3681	0.3678	0.3692	0.3577	0.3190	0.3312	0.3771
Simple								
	No Text	Initialization			Alignment			Augmentation
		Name	Desc.	Syn.	Name	Desc.	Syn.	
category	0.0461	0.3591	0.3536	0.3668	0.0520	0.3263	0.2619	0.3367
ATC	0.3278	0.3820	0.3617	0.3732	0.3644	0.3410	0.3425	0.3475
pathway	0.7513	0.7164	0.7299	0.7180	0.7336	0.7428	0.7448	0.7189
interact	0.6215	0.7229	0.7253	0.7338	0.6488	0.6242	0.6602	0.7230
target	0.0815	0.1128	0.1169	0.1171	0.0971	0.0918	0.0873	0.1163
enzyme	0.1903	0.2442	0.2143	0.2555	0.2499	0.2031	0.1977	0.2304
carrier	0.1358	0.2544	0.2441	0.2526	0.1881	0.1766	0.1266	0.1493
transporter	0.2242	0.2718	0.2189	0.2543	0.2396	0.2042	0.2417	0.2173
Avg. (Macro)	0.2973	0.3829	0.3705	0.3839	0.3216	0.3387	0.3328	0.3549

Table 6.4: Comparison of MRR performance for each method. The MRR for each relational triple and calculated the macro-averaged MRR are summarized. The highest score for each node row is shown in bold.

pathway, but on the categorical relation *category* and *ATC*, the MRR was improved when the Initialization method is adopted. Initialization methods that use Name information improved the MRR of *target*, *enzyme*, *carrier* and *transporter*, which are the relations

between drugs and proteins. The averaged MRR was lower than that of the models without textual information.

When the ComplEx scoring function is used, the MRR decreased in the *interact* and *pathway* relation, while the MRR increased on the categorical relations and relations between drugs and proteins, these are the same tendency as the DistMult algorithm. Especially in the *category* relation, the ComplEx scoring function model without text information has a much lower MRR than TransE or DistMult-based models, but the performance was improved by using text information. The Initialization and Augmentation methods show higher macro-averaged MRR than the model without text information.

When the SimplE scoring function is used, the model without text information showed the lowest macro-averaged MRR; however, the Initialization model that used the Synonyms information showed a higher MRR than the model without text information for all relation types except *pathway*, and showed the highest macro-averaged MRR in all models. These results showed that it is effective to utilize text information during updating KG embeddings under the SimplE scoring function.

These results show that the utility of textual information for learning KG embeddings depends on the scoring functions and relation types. The textual information is always useful in predicting categorical relations such as *category* and *ATC*, while the text information can be harmful for other relations and the utility depends on the scoring functions. The best settings for each relation type are summarized in Table 6.5. This shows that there is no best single embedding method. The best method to incorporate text information including No Text and the most useful text type also depends on the relation types.

6.6.1 Analysis of the Data Ombalance of the Constructed KG

Why some models that use text information show lower performance in *interact* and *pathway* relation and show higher performance in categorical relation and drug-protein

	MRR	Method	Text Information
category	0.3668	SimplE	Initialization+Synonyms
ATC	0.3820	SimplE	Initialization+Name
pathway	0.7513	SimplE	No Text
interact	0.8678	ComplEx	No Text
target	0.1243	ComplEx	Augmentation
enzyme	0.3590	TransE	Alignment+Synonyms
carrier	0.4155	TransE	No Text
transporter	0.3576	TransE	No Text
Avg. (Macro)	0.3839	SimplE	Initialization+Synonyms

Table 6.5: Summary of the best settings for each relation

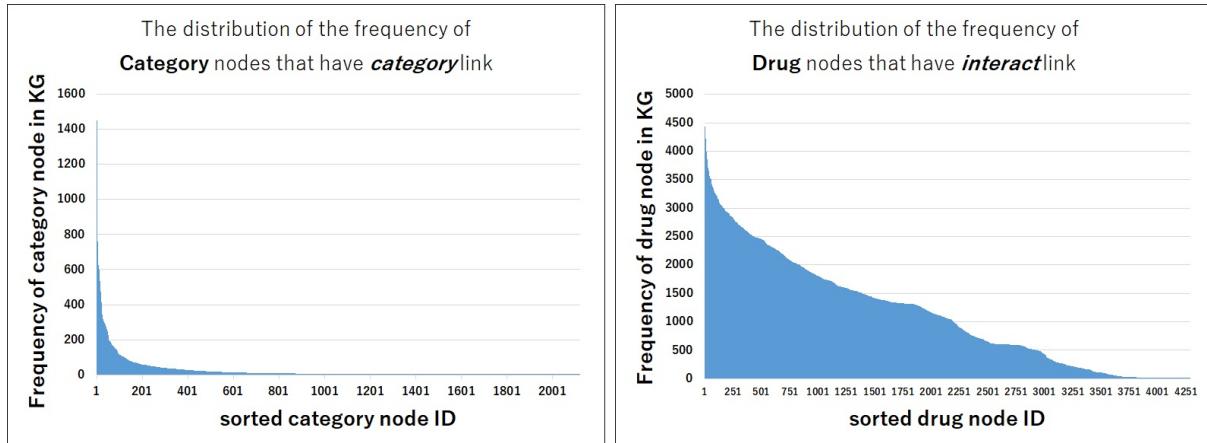


Figure 6.3: The distribution of the frequency of nodes in the train data set. The frequencies of category nodes linked by *category* relation are highly imbalanced, while the frequencies of drug nodes linked by *interact* relation are less imbalanced.

relation? In order to analyze these tendencies, The frequency of nodes in the constructed KG is investigated. Figure 6.3 shows the distribution of the frequencies of category nodes that have the *category* link and drug nodes that have the *interact* link in train triples. Compared with the distribution of drug nodes frequency, the frequency distribution of category nodes is extremely imbalanced. The distribution shows that a small part of category nodes have a large number of triples between drugs, and many other category nodes have few triples, it could be difficult to predict triples that contain these nodes. Even if it is difficult to train the representation of nodes from the information of KG structure, it may be possible to predict the correct triples by utilizing the textual embeddings encoded by pre-trained BERT.

	Averaged MRR
Full text nodes	0.3771
- description	0.3626
- synonyms	0.3655
- indication (drug)	0.3761
- pharmacodynamics (drug)	0.3727
- mechanism-of-action (drug)	0.3646
- metabolism (drug)	0.3689
- gene-name (protein)	0.3754

Table 6.6: Ablation study of text information on Augmentation method (ComplEx score function)

6.6.2 Ablation Study of Augmentation Method

In the Augmentation method, multiple text items can be considered at the same time.

Table 6.6 shows the results of removing each text item. Here, description and synonyms are text items that heterogeneous entities have in common, indication, pharmacodynamics, mechanism-of-action and metabolism are text items that only drug entities have, and gene-name is that only protein entities have. From Table 6.6, it can be seen that the averaged MRR becomes lower regardless of which text items are removed, and these results show that all text items are effective for the link prediction task. In addition, the averaged MRR drops greatly when description or synonyms is excluded, these are the text items that many entities have. The averaged MRR also drops greatly when text information with high coverage is excluded, such as metabolism-of-action.

6.6.3 Effect of Node Type Filtering

As explained in Sections 6.3.3, 6.5.4, 6.5.6, the model filters out impossible negative instances by restricting the types of entities in the relations. Table 6.7 shows the effect of the entity type filtering. Overall, by performing entity type filtering, the averaged MRR is improved. In particular, in the Augmentation method, entity type filtering is very effective; this is because the Augmentation method adds textual nodes to the graph and is more likely to create an inappropriate negative example during negative sampling.

		No Text	Initialization			Alignment			Aug.
			Name	Desc.	Syn.	Name	Desc.	Syno.	
TransE	w/ type filtering	0.3319	0.2930	0.2950	0.3015	0.3337	0.3318	0.3302	0.2883
	w/o type filtering	0.2759	0.2373	0.2265	0.2429	0.2738	0.2722	0.2753	0.1932
DistMult	w/ type filtering	0.3380	0.3206	0.2979	0.3162	0.3261	0.3274	0.3265	0.3011
	w/o type filtering	0.2217	0.2285	0.2423	0.2547	0.2462	0.2219	0.2464	0.1449
ComplEx	w/ type filtering	0.3242	0.3681	0.3678	0.3692	0.3577	0.3190	0.3312	0.3771
	w/o type filtering	0.2848	0.2906	0.2887	0.2981	0.2931	0.3052	0.3095	0.2373
SimplE	w/ type filtering	0.2973	0.3829	0.3705	0.3839	0.3216	0.3387	0.3328	0.3549
	w/o type filtering	0.2848	0.2906	0.2887	0.2981	0.2931	0.3052	0.3095	0.2373

Table 6.7: Comparison of averaged MRR performance for w/ (with) and w/o (without) entity type filtering

6.6.4 Case Study

As can be seen from Table 6.5, textual information acts harmfully in some relations. In this section, the content of the text is analyzed to investigate when the text information is harmful or helpful. Table 6.8 shows examples of improved or worsened score ranks on the link prediction task. The examples are where the difference between the rank of the textual model and the rank of the non-textual model is largest, that is, examples where textual information is most useful or harmful for each relation type. In addition, the cases are narrowed down where the better rank is 1. In example (a), the highlighted “*stereoisomers*” in the description of the drug entity appears in the synonyms of the category entity. Similarly, in example (b), “*antiviral*” in the description of the drug entity appears in the name of the ATC entity. The description of the drug entity directly mentions the category in which the drug is included, which is thought to have helped to predict the link of the categorical relation type.

On the other hand, for the examples where the textual information is most harmful, in example (c), the description of protein “*Cytochrome P450 2J2*” does not directly mention the “*Etoricoxib Action Pathway*” pathway. In example (d), the description of each drug entity mainly describes the indication of the drug, not the relationship to other drugs. It is difficult to tell the cause of the poor rank because multiple factors may be involved, but the description of the head entity mainly explains the function and role of the head entity itself, and there is no description that mentions the relationship with the tail entity. This

point is considered to be one of the causes of the textual information becoming noise.

6.7 Summary

A new heterogeneous knowledge graph containing textual information PharmaHKG from several databases is constructed. The combinations of three methods to use textual information and four scoring functions on the link prediction task are compared. The utility of text information and the best combination for the link prediction depend on the target relation types. In addition, when the averaged MRR is focused on all relation types, a method that combines SimplE and text information achieved the highest MRR, and this result showed the usefulness of text information in the link prediction task in the drug domain.

Examples where textual information is **helpful**

(a) Relation: *category*, textual model rank:1, non-textual model rank:65

Head	ID	DB13746 (drug entity)
	Name	<i>Bioallethrin</i>
	Desc.	<i>Bioallethrin refers to a mixture of two of the allethrin isomers (1R,trans;1R and 1R,trans;1S) in an approximate ratio of 1:1, where both isomers are active ingredients. A mixture of the two same stereoisomers, but in an approximate ratio of R:S in 1:3, is called esbiothrin.</i>
	Syn.	<i>Depalléthrine</i>
Tail	ID	D013237 (category entity)
	Name	<i>Stereoisomerism</i>
	Desc.	<i>The phenomenon whereby compounds whose molecules have the same number and kind of atoms and the same atomic arrangement, but differ in their spatial relationships.</i>
	Syn.	<i>Molecular Stereochemistry, Stereochemistry, Molecular, <u>Stereoisomers</u>, <u>Stereoisomer</u></i>

(b) Relation: *ATC*, textual model rank:1, non-textual model rank:25

Head	ID	DB00369 (drug entity)
	Name	<i>Cidofovir</i>
	Desc.	<i>Cidofovir is an injectable antiviral medication employed in the treatment of cytomegalovirus (CMV) retinitis in patients diagnosed with AIDS.</i>
	Syn.	<i>CDV, Cidofovir anhydrous, Cidofovirum</i>
Tail	ID	J05A (ATC entity)
	Name	<i>DIRECT ACTING ANTIVIRALS</i>
	Desc.	ATC entity has no description
	Syn.	ATC entity has no synonyms

Examples where textual information is **harmful**

(c) Relation: *pathway*, textual model rank:25, non-textual model rank:1

Head	ID	P51589 (protein entity)
	Name	<i>Cytochrome P450 2J2</i>
	Desc.	<i>This enzyme metabolizes arachidonic acid predominantly via a NADPH-dependent olefin epoxidation to all four regioisomeric cis-epoxyeicosatrienoic acids.</i>
	Syn.	<i>1.14.14.1, Arachidonic acid epoxygenase, CYPIIJ2</i>
Tail	ID	SMP0000695 (pathway entity)
	Name	<i>Etoricoxib Action Pathway</i>
	Desc.	<i>Etoricoxib (also named as Arcoxia) is a COX-2 selective inhibitor. It can be used to treat fever, pain, swelling, inflammation, and platelet aggregation.</i>
	Syn.	pathway entity has no synonyms

(d) Relation: *interact*, textual model rank:4,119, non-textual model rank:1

Head	ID	DB08893 (drug entity)
	Name	<i>Mirabegron</i>
	Desc.	<i>Mirabegron is a beta-3 adrenergic receptor agonist for the management of overactive bladder. It is an alternative to antimuscarinic drugs for this indication.</i>
	Syn.	<i>Mirabegron</i>
Tail	ID	DB00937 (drug entity)
	Name	<i>Diethylpropion</i>
	Desc.	<i>A appetite depressant considered to produce less central nervous system disturbance than most drugs in this therapeutic category. It is also considered to be among the safest for patients with hypertension.</i>
	Syn.	<i>alpha-Benzoyltriethylamine, alpha-Diethylaminopropiophenone, Amfepramone</i>

Table 6.8: The content of the text in the examples where the difference between the rank of textual model and the rank of non-textual model is largest for each relation type. The score function SimpLE was used for *category*, *ATC* and *pathway* relation and ComplEx was used for *interact* relation. The Augmentation model was selected as the model with textual information. The highlighted part is the mention common to head and tail entities. The Description and Synonyms are partly excerpted due to space limitations.

7 Integrating Heterogeneous Domain Information for Relation Extraction

This chapter proposes a novel method that utilizes the heterogeneous KG information for relation extraction. This chapter includes work from Asada et al. (2022) [11].

7.1 Background

Chapter 6 integrated multiple databases into a heterogeneous KG and conducted a link prediction task on the heterogeneous KG to obtain representation vectors of the drugs.

This chapter proposes a novel method that effectively combines the input sentence information and the heterogeneous KG information to extract DDIs from texts. The model on the data set of the DDIExtraction-2013 task is evaluated to demonstrate the usefulness of heterogeneous KG information.

In Chapter 5, the approach to combining the input sentence representation vector by BERT with the description and molecular structure representation vectors was simply concatenating the respective representation vectors. This approach is considered insufficient to capture the correlation between the context around the drug mentions in the input sentence and the external knowledge information.

This chapter uses the idea of “entity marker” to devise a model that incorporates the heterogeneous KG embeddings from the lowest layer of BERT and integrally considers the correlation of input sentence information and heterogeneous KG information. Many relation extraction methods based on the entity marker idea have been studied [97, 98]. The levitated marker model [99] that most inspired the proposed method is described. Figure 7.1 shows the overview of solid marker [100, 101] and levitated marker [99]. Solid marker explicitly inserts two solid markers before and after the span to highlight the span in the input sentence. Here, [Md] and [TK] stand for Method and Task, and these markers indicate the type of mentions in the sentence. The levitated marker first sets the pair of markers to share the same position with the target tokens and then ties a pair

Solid Marker

[Md] **MORPA**[/Md] is a fully implemented [Md]**parser**[/Md] for a [Tk]**text - to - speech**[/Tk] system.

Levitated Marker

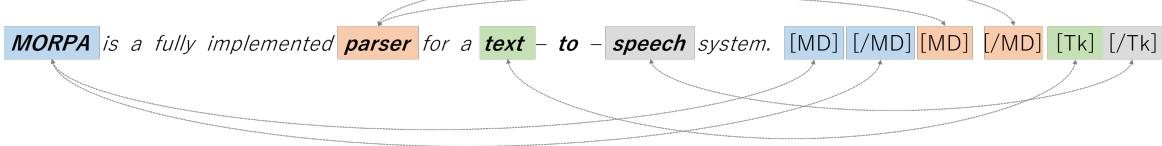


Figure 7.1: Overview of the levitated marker. Tokens of the same color share the position IDs in the levitated marker.

of markers by a directional attention. The levitated marker can identify the mentions without collapsing the original input sentence. The mention and the marker are tied by an attention mechanism of the Transformer model. The levitated marker embeddings are replaced with the KG embedding to which the target drug mention corresponds, and consider the relationship between the word embeddings and the KG embeddings for DDI extraction.

7.2 Method

7.2.1 Obtaining Heterogeneous KG Embeddings

In constructing the heterogeneous KG, the augmentation method described in Chapter 6 is used. In the Augmentation method, textual nodes as well as entity nodes on the heterogeneous KG are placed, and the relational triples in the train data set are augmented.

One extension is made from the heterogeneous KG of Chapter 6, that is, the molecular structural nodes of the drugs are added to the heterogeneous KG. An overview of the newly constructed KG is shown in Figure 7.2. Similar to initializing textual nodes with embeddings by a pre-trained BERT model, molecular structural nodes with a pre-trained model of the SMILES string coding representation embeddings [102] are also initialized.

The constructed pharmaceutical heterogeneous KG enables us to obtain Drug representation vectors that take into account various information such as hierarchical categorical information, interacted protein information, related pathway information, drug

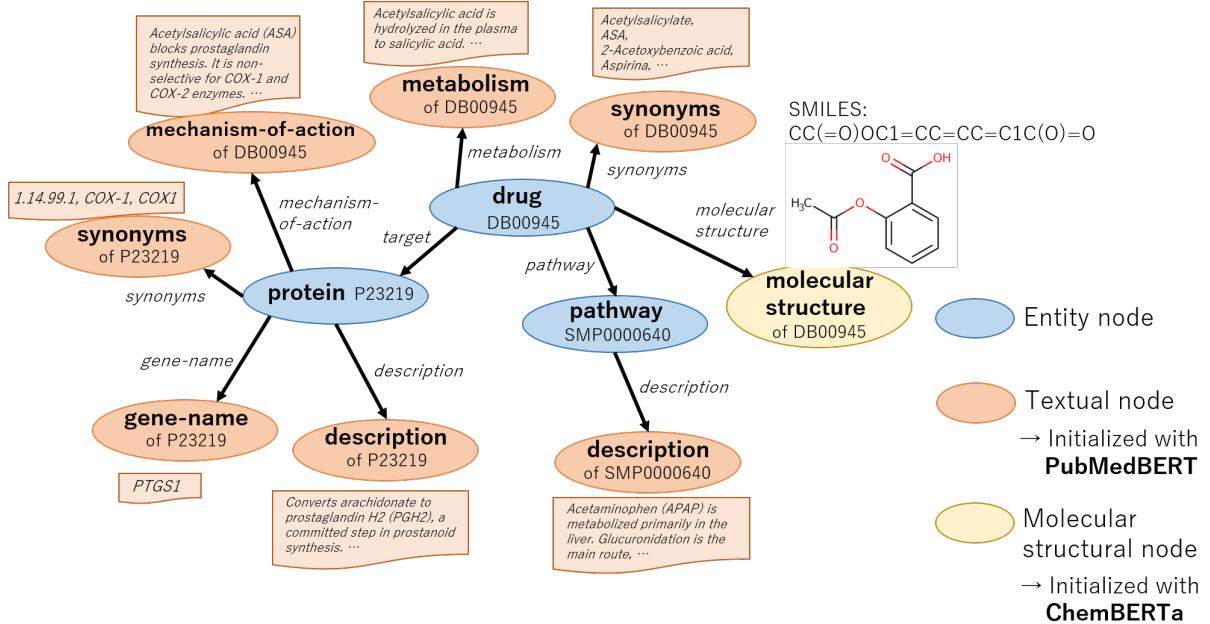


Figure 7.2: Heterogeneous KG with additional drug molecular structure information

textual information, and drug molecular structural information. In the next section, a novel method for DDI extraction from the literature using the obtained heterogeneous KG representations of drugs is described.

7.2.2 DDI Extraction from Texts with Heterogeneous KG Embeddings

An overview of the DDI extraction model is shown in Figure 7.3. the proposed model adopts the idea of a levitated entity marker, and two drug mention markers are placed at the end of a sentence.

Embedding Layer The input sentence S is tokenized to sub-word units by the BERT tokenizer and converted into the format shown below:

$$S = \{ [\text{CLS}], w_1, w_2, \dots, w_{m_1}, \dots, w_{m_2}, \dots, [\text{SEP}], [\text{KG1}], [\text{KG2}] \}, \quad (7.1)$$

where w_i is the i -th sub-word, and $[\text{CLS}]$, $[\text{SEP}]$ are the special tokens of BERT, m_1 is the drug mention 1 ($DRUG1$), and m_2 is the drug mention 2 ($DRUG2$), and $[\text{KG1}]$, $[\text{KG2}]$ are markers for mapping mentions and KG entries.

Then, in the lowest embedding layer of the BERT model, the sub-word w_i and special tokens are looked up from the pre-trained BERT embedding table and converted to

embedding vectors. In addition, the marker embeddings are replaced with the heterogeneous KG embeddings. All tokens are converted to embedding vectors and the embedding matrix \mathbf{W}^0 of the input sentence is shown as follows:

$$\mathbf{W}^0 = \{\mathbf{w}_{\text{CLS}}, \mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{m_1}, \dots, \mathbf{w}_{m_2}, \dots, \mathbf{w}_{\text{SEP}}, \mathbf{w}_{\text{KG1}}, \mathbf{w}_{\text{KG2}}\}, \quad (7.2)$$

Here, \mathbf{w}_i , \mathbf{w}_{CLS} and \mathbf{w}_{SEP} are looked up from the BERT embedding table $\mathbf{V}_{BERT} \in \mathbb{R}^{N_v \times d}$ and \mathbf{w}_{KG1} , \mathbf{w}_{KG2} are looked up from the heterogeneous KG embedding table $\mathbf{V}_{KG} \in \mathbb{R}^{N_e \times d}$. d is the dimension of the embedding vector, and N_v is the number of vocabularies of the BERT tokenizer, and N_e is the number of entities in the heterogeneous KG.

Self-Attention Layer The embedding matrix \mathbf{W}^0 is the input to the L -layers of BERT self-attention module:

$$\mathbf{W}^{l+1} = \text{SelfAttention}^l(\mathbf{W}^l), \quad (7.3)$$

where $l = 0, 1, 2, \dots, L - 1$. The output of the final attention layer \mathbf{W}^L is shown as follows:

$$\mathbf{W}^L = \{\mathbf{h}_{\text{CLS}}, \mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_{m_1}, \dots, \mathbf{h}_{m_2}, \dots, \mathbf{h}_{\text{SEP}}, \mathbf{h}_{\text{KG1}}, \mathbf{h}_{\text{KG2}}\}, \quad (7.4)$$

where $\mathbf{h}_i \in \mathbb{R}^d$ is the hidden state vector of i -th token.

As shown in Figure 7.3, mention 1 and its KG entity, and mention 2 and its KG entity share the position ID, which ties mention and marker by directional attention.

Prediction Layer The loss function is calculated from the hidden representation vectors of the final layer of BERT architecture. First, the hidden representation of the [CLS] token and two drug mention tokens are concatenated as follows:

$$\mathbf{h}_{all} = [\mathbf{h}_{\text{CLS}}; \mathbf{h}_{m_1}; \mathbf{h}_{m_2}]. \quad (7.5)$$

The concatenated representation vector \mathbf{h}_{all} is passed through a dense layer and middle layer representation is obtained,

$$\mathbf{h}_{mid} = \mathbf{W}_{mid}\mathbf{h}_{all} + \mathbf{b}_{mid}, \quad (7.6)$$

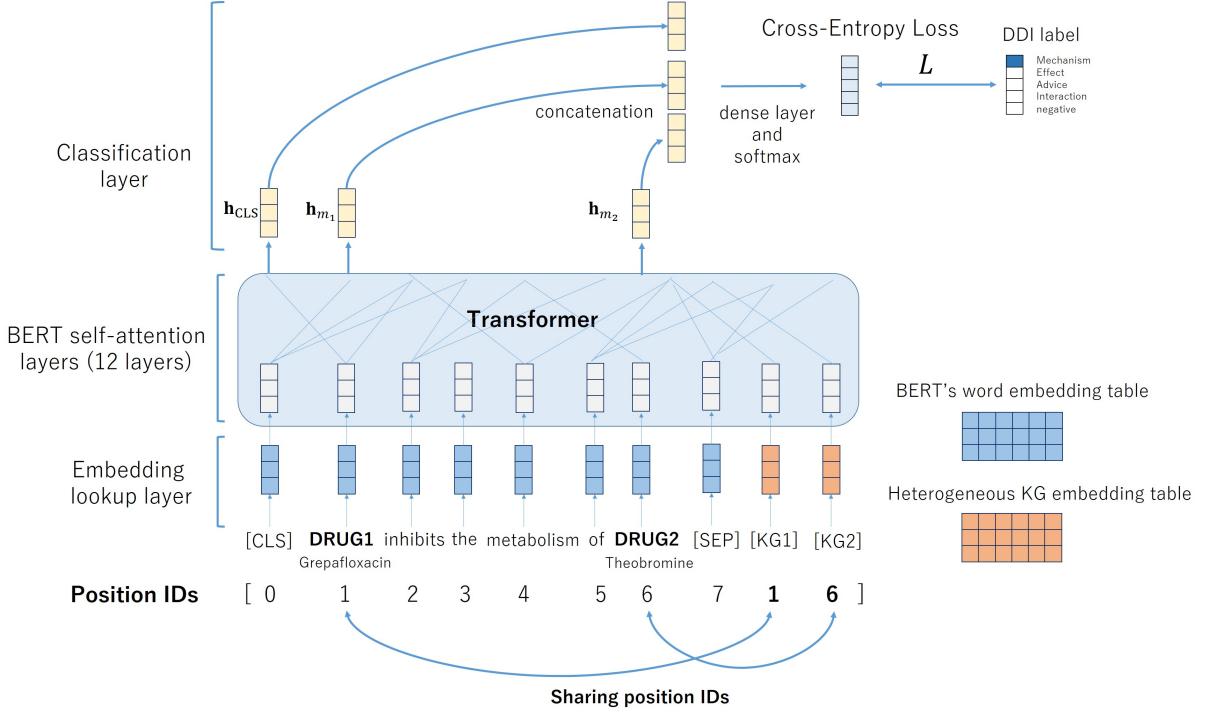


Figure 7.3: The DDI extraction model which utilize heterogeneous information of drugs

where $\mathbf{W}_{mid} \in \mathbb{R}^{3d \times d_m}$, $\mathbf{b}_{mid} \in \mathbb{R}^{d_m}$ are the trainable weight and bias, and d_m is the dimension of middle layer vector. Then the middle layer representation is converted into fully-connected representation as follows:

$$\mathbf{h}_{fc} = \mathbf{W}_{fc}\mathbf{h}_{mid} + \mathbf{b}_{fc}, \quad (7.7)$$

where $\mathbf{W}_{fc} \in \mathbb{R}^{d_m \times c}$, $\mathbf{b}_{fc} \in \mathbb{R}^c$ are the trainable weight and bias, and c is the number of label types. The fully-connected representation vector \mathbf{h}_{fc} is converted to probability form by the softmax function:

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{h}_{fc}). \quad (7.8)$$

The cross-entropy loss between the prediction probability $\hat{\mathbf{y}}$ and the gold label \mathbf{y} is employed,

$$L = - \sum \mathbf{y} \log \hat{\mathbf{y}}. \quad (7.9)$$

The model parameters are updated to minimize the loss L .

7.3 Experimental Settings

7.3.1 Mention Linking

In contrast to the previous chapters, in this chapter, not only with DrugBank drug entities but also with MeSH categorical terms, ATC code categorical terms, and UniProt protein terms are linked. The DDIExtraction-2013 shared task data set consists of four types of entities, *DRUG*, *DRUG.N*, *BRAND*, and *GROUP*. The *GROUP* type drug mentions may be linked to categorical terms, so the linking coverage is better than in the previous chapters.

As a result, 97.05% of the unique mentions in train data set were linked to heterogeneous KG entries, and 97.89% of the unique mentions in test data set were linked. As for the coverage on instances where mention 1 and mention 2 are both linked, the coverage of train data instances was 91.90% (25,540 / 27,792), and the coverage of test instances was 90.75% (5,187 / 5,716).

When the drug mention is not linked to the KG entries, the special tokens (KG1 or KG2) are masked and these embeddings are excluded from the attention calculation. In this way, the proposed model takes into account KG embedding information for linked examples, and for unlinked examples, the model behaves like the baseline model.

7.3.2 Link Prediction Settings

The same train/validation/test split triples is used as the data sets created in Chapter 6 and the link prediction task is conducted. Mini-batch training using the Adagrad [103] optimizer is employed.

Hyper-parameter tuning is conducted on the validation data set. Hyper-parameters include initial learning rate and mini-batch size. As in Chapter 6, TransE, DistMult, ComplEx and SimplE were used for score functions. The employed hyper-parameters is shown in Table 3.3.

SMILES strings are extracted from DrugBank database. The 9,859 drug entities in

the heterogeneous KG data set have SMILES strings. Relation triples (drug, *structure*, SMILES) are added to the train data set and molecular structural nodes (SMILES nodes) are initialized by the embedding vectors of the pre-trained SMILES representation language model.

The [CLS] token representation of PubMedBERT [93] is used as the initial value of the textual nodes, and ChemBERTa [102] was used as the initial value for the molecular structural nodes. Chrithranada et al. pre-trained ChemBERTa on 77M unique SMILES from PubChem [104], the world’s largest open-source collection of chemical structures. The SMILES were canonicalized and globally shuffled to facilitate large-scale pre-training. ChemBERTa is based on the RoBERTAa [105] transformer model. In pre-training, ChemBERTa model masks 15% of the tokens in each SMILES string.

7.3.3 DDI Extraction Settings

The AdamW optimizer [58] is employed, and mixed-precision training [59] is employed for memory efficiency. The weight averaging [106, 107] technique is employed, where all model parameters are saved at each update and the model predicts the DDI label from the average of all stored parameters.

PubMedBERT is employed as the textual representation model for the DDI extraction task. The word embeddings of PubMedBERT and heterogeneous KG embeddings are frozen during training.

Stratified 5-fold cross-validation is conducted on the DDIEtraction-2013 train data set for hyper-parameter tuning. Hyper-parameters include learning rate, weight decay coefficient, dropout probability and mini-batch size. The employed hyper-parameters are shown in Table 7.1. The significance results are based on the Randomization test [51].

7.4 Results and Discussions

First, the results of link prediction task on the heterogeneous KG are shown in Table 7.2. For each of the four score functions, TransE, DistMult, ComplEx and SimplE, the four

Parameter	Value
Link Prediction	
Entity embedding size	768
Learning rate	0.25
Regularization parameter	9e-9
Mini-batch size	8,192
DDI Extraction	
Embedding size	768
Middle layer size	768
Maximum sequence length	256
Learning rate	5e-5
Regularization parameter	8e-0
Dropout probability	0.5
Mini-batch size	128

Table 7.1: Hyper-parameters for link prediction and DDI extraction model

methods listed below are evaluated:

entity nodes only This is a method that trains only from the nodes contained in the heterogeneous KG. In Figure 7.2, only the actual nodes (the blue one) is included in the train data set.

with textual nodes In this method, in addition to the actual nodes, pseudo nodes that hold textual information such as synonyms and descriptions of entity items are added to the heterogeneous KG. In Figure 7.2, textual nodes (the red one) are added to the actual nodes.

with molecular structural nodes In this method, pseudo molecular structure modes are added to the heterogeneous KG in addition to the actual nodes. As shown in Figure 7.2, molecular structural nodes (the yellow ones) are added.

with textual nodes and molecular structural nodes In this method, both textual nodes and molecular structural nodes are added. This approach can consider a wide variety of heterogeneous information about drugs.

Table 7.2 showed that the TransE model performs poorly for both MRR and Hits@k. This should be due to the inability of the TransE model to capture the symmetrical relational triples. The TransE model showed low performance of MRR and Hits@k because the

	TransE	DistMult	ComplEx	SimplE
entity nodes only				
MRR	0.3114	0.6732	0.6627	0.6228
Hits@1	0.0108	0.5416	0.5436	0.3954
Hits@3	0.5590	0.7680	0.7377	0.8287
Hits@10	0.7364	0.9138	0.8892	0.9481
with textual nodes				
MRR	0.2894	0.7702	0.7874	0.7175
Hits@1	0.0092	0.6199	0.6424	0.5019
Hits@3	0.5102	0.9104	0.9258	0.9303
Hits@10	0.6987	0.9703	0.9722	0.9744
with molecular structural nodes				
MRR	0.3003	0.7677	0.7313	0.7156
Hits@1	0.0094	0.6171	0.5383	0.4987
Hits@3	0.5352	0.9092	0.9180	0.9307
Hits@10	0.7195	0.9700	0.9717	0.9746
with textual nodes and molecular structural nodes				
MRR	0.2877	<u>0.7933</u>	0.7923	0.7235
Hits@1	0.0091	<u>0.6610</u>	0.6509	0.5086
Hits@3	0.5051	0.9166	0.9279	<u>0.9386</u>
Hits@10	0.6995	0.9711	0.9729	<u>0.9753</u>

Table 7.2: The comparison of link prediction performance on heterogeneous KG. Figures marked in bold indicate the highest performance when limited to each individual score function. The underlines indicate the highest performance for all score functions.

heterogeneous KG contained a large proportion of the (drug, *interact*, drug) triples, which is a symmetric relationship. Furthermore, the TransE model showed the highest MRR and Hits@k when using the “entity nodes only” method, meaning that adding textual or molecular structural nodes did not improve link prediction performance.

On the other hand, DistMult, ComplEx, and SimplE, which can consider symmetric relationships, showed higher performance than TransE. These models successfully improved the performance of link prediction task by adding textual nodes and molecular structural nodes, respectively. And when both textual nodes and molecular structural nodes are added, the further performance improvement was achieved. As shown by the underlined values in Table 7.2, the highest performance for all MRR and Hits@k metrics was achieved by the method using both textual and molecular structural nodes. These results show that rich embedding representations are obtained by considering various heterogeneous domain information.

	Method	P	R	F (%)
Reported scores	CNN [26]	75.29	60.37	67.01
	BiLSTM [31]	67.77	66.80	67.28
	PubMedBERT [93]	-	-	82.42
	SciFive-Large [108]	-	-	83.67
The author’s implementation	CNN + Mol. [43]	73.31	71.81	72.55
	SciBERT + Mol. [9]	83.57	82.12	82.84
	SciBERT + Desc. [9]	84.05	81.81	82.91
	SciBERT + Mol. + Desc. [9]	85.36	82.83	84.08
	PubMedBERT (baseline)	83.45	83.96	83.70
	PubMedBERT + HKG	85.32	85.49	85.40*

Table 7.3: The comparison of DDI extraction performance on DDIExtraction-2013 test data set. * indicates performance improvement from PubMedBERT (baseline) at a significance level of $p < 0.001$.

Then, the performance of DDI extraction models that leverage these heterogeneous KG embeddings is described. The score functions and hyper-parameters are chosen by the results of 5-fold cross-validation, as described in later sections. Table 7.3 shows the performances evaluated on the DDIExtraction-2013 task test set. The proposed model PubMedBERT+HKG achieved the micro-averaged F-score of 85.40%, showing the current state-of-the-art performance. The proposed model achieved a significant F-score improvement of 1.70 percent points over the baseline model by using heterogeneous information about drugs. Compared to other existing models, the PubMedBERT+HKG model showed a higher F-score. The SciBERT+Mol.+Desc. model is an ensemble of SciBERT+Mol. and SciBERT+Desc. The proposed model showed higher performance than the ensemble of multiple models.

Then, the F-scores for each of the four DDI labels of DDIExtraction-2013 task data set are shown in Table 7.4. As shown in Table 7.4, the model with heterogeneous KG information improves F-scores for all DDI types. In particular, the F-score greatly improves for *Mechanism* relation. HKG model improved F-score by 3.80 percent points from the baseline model.

Method		Mech.	Effect	Adv.	Int.	Avg. (%)
PubMedBERT (baseline)	P	87.12	87.18	78.71	82.69	83.93
	R	85.10	92.31	88.33	44.79	77.63
	F	86.10	89.67	83.25	58.11	79.28
PubMedBERT + HKG	P	88.96	88.84	81.06	81.13	84.99
	R	88.08	93.67	89.17	44.79	78.92
	F	88.52	91.19	84.92	57.72	80.58

Table 7.4: The comparison of F-scores for individual DDI types and macro-averaged F-score on DDIExtraction-2013 test data set

Method	P	R	F (%)
baseline	82.19	83.23	82.54
+ HKG (TransE)	83.46	84.37	83.82
+ HKG (DistMult)	83.68	85.42	84.48
+ HKG (ComplEx)	83.68	84.32	83.90
+ HKG (SimpleE)	83.46	84.50	83.86

Table 7.5: The comparison of DDI extraction performances with different score functions for training KG embeddings. the author performs 5-fold cross-validation and show the average of F-scores over the five validation data sets.

7.4.1 Selecting Score Functions

In this section, the author will discuss which score function was effective for DDI extraction. Table 7.3 shows the average F-scores for the five validation data sets for each score function. F-score is higher than the baseline model when using heterogeneous KG embeddings trained by any of score functions. The improvement of F-score points from the baseline model is 1.28 for the TransE model and 1.94 for the DistMult model and 1.36 for the ComplEx and 1.32 for the SimpleE model. From these results, the DistMult score function is adopted for the DDI extraction model. The DistMult model preformed best on the link prediction task on MRR, Hits@1, and also showed the best performance on the DDI extraction task.

7.4.2 Ablation Study on Model Architecture

This section provides ablation studies. Table 7.6 shows the ablation study results on 5-fold cross-validation data sets.

w/o Sharing Position IDs First, the case excluding the sharing of position IDs is discussed. The sharing of position IDs has the effect of linking the mention embeddings and

Method	P	R	F (%)	Δ (pp)
Full model	83.68	85.42	84.48	-
w/o sharing position ids	82.49	86.04	84.18	0.30
w/o freezing KG embeddings	83.69	84.32	83.90	0.58
w/o CLS representation	82.63	85.17	83.81	0.67
w/o mention representation	82.07	85.69	83.78	0.70
w/o KG embeddings (baseline)	82.19	83.23	82.54	1.94

Table 7.6: The ablation study on the model architecture. The performance with 5-fold cross-validation on the training set is showed.

KG embeddings. As shown in Figure 7.3, with position sharing, the position ID of [KG1] is 1 of the drug mention 1 and the position ID of [KG2] is 6 of the drug mention 2. When this sharing is disabled, the position IDs of [KG1] and [KG2] are the values following from the IDs of the [SEP] token. From Table 7.6, when position sharing is excluded, the F-score is reduced by 0.30 percent points from the full model.

w/o freezing KG embeddings In the proposed model, the BERT embeddings the KG embeddings are frozen and the attention weight is trainable. Table 7.6 shows that when embedding freezing is disabled, 0.58 percent points of F-score is lower than the full model. The author thinks the reason embedding freezing is effective is that if the embedding is not frozen, there will be a gap between the KG embeddings when drugs that appear on the train set and those that appear only on the test set.

w/o CLS representation When \mathbf{h}_{CLS} was excluded from the input vector of the middle layer, the F-score decreased by 0.67 percent points. The CLS token representation holds information of the entire sentence. It is effective to use the representation of the [CLS] token.

w/o mention representation When \mathbf{h}_{m_1} and \mathbf{h}_{m_2} were excluded from the input vector of the middle layer, the F-score decreased by 0.70 percent points. In addition to the [CLS] token representation, it is effective to use the drug mention representation.

Method	P	R	F (%)	Δ (pp)
PubMedBERT + HKG	83.68	85.42	84.48	-
w/o Protein nodes	83.22	85.04	84.08	0.40
w/o MeSH category nodes	83.28	83.75	83.38	1.10
w/o ATC nodes	83.33	84.27	83.73	0.75
w/o Pathway nodes	83.98	84.85	84.30	0.18
w/o Textual nodes	83.75	84.20	83.86	0.62
w/o Molecular structure nodes	83.36	84.30	83.71	0.77

Table 7.7: The ablation study on node types. The performance with 5-fold cross-validation on the training set is showed.

w/o KG embeddings Finally, the case without KG embeddings is discussed. In the baseline model, [KG1] and [KG2] tokens are not fed to the BERT architecture and KG embeddings are not used. Except for this point, the model structure is the same as the proposed model.

The use of heterogeneous KG information increased the F-score by 1.94 percent points. This result showed that the drug heterogeneous information is useful for DDI extraction from literature.

7.4.3 Ablation Study on Heterogeneous KG Node Types

Table 7.7 shows the ablation study on the effect of individual KG node type. As shown in Table 7.7, all types of nodes in the heterogeneous KG contribute to the performance improvement of DDI extraction from the literature. The results show the importance of simultaneously considering multiple pieces of drug-related information. Among the types of nodes, the MeSH categorical information contributed most to the performance improvement while the pathway information contributed least.

7.4.4 Verification of DDI Label Leakage from KGs

The constructed heterogeneous KG contains triples of *interact* relations between Drug nodes and Drug nodes. This section examines whether the *interact* triples directly affects the performance improvement in DDI extraction from the literature and whether there is a leakage of DDI labels. First, the percentage of drug pairs in the DDIEtraction-2013 train data set that are registered as *interact* relations in the KG is shown. 39.79% (11,060 /

Method	P	R	F (%)
All drug pairs			
baseline	82.19	83.23	82.54
baseline + HKG	83.68	85.42	84.48
Drug pairs that are included in KGs as <i>interact</i> triples			
baseline	85.11	85.74	85.25
baseline + HKG	85.94	86.73	86.22
Drug pairs that are NOT included in KGs as <i>interact</i> triples			
baseline	79.83	80.94	80.23
baseline + HKG	81.83	84.32	83.02

Table 7.8: The DDI extraction performance for drug pairs that are included in the constructed heterogeneous KG and drug pairs that are not included in KG

27,792) of the drugs pairs are included in the constructed KG and others are not included in the KG. It should be noted that the DDI labels in the DDIExtraction-2013 data set are *Mechanism*, *Effect*, *Advice* and *Int.*, whereas there is only one relation label *interact* in the KG. Table 7.8 shows a comparison of DDI extraction performance for drug pairs that are included in the KG and drug pairs that are not included in the KG. As shown in Table 7.8, both for drug pairs that are registered in the KG and drug pairs that are not registered in the KG, the proposed model (baseline + HKG) outperformed the baseline model. When the pairs are included in the KG, the improvement of F-score is 1.1 percent, 0.97 pp, whereas when the pairs are not included in the KG, the improvement of F-score is 3.4 percent, 2.79 pp. Therefore, the performance improvement of the proposed model is greater when the drug pairs is not registered in the KG than when they are registered in the KG. The author believes that these results indicate that the proposed model does not improve DDI extraction performance because of the leakage of DDI labels from the constructed KG.

7.4.5 Learning Curve

The learning curve of baseline model and proposed model is shown in Figure 7.4. Figure 7.4 plots the average of the F-scores of the five validation data sets per epoch. Learning curves show that the proposed method always has a higher F-score than the baseline

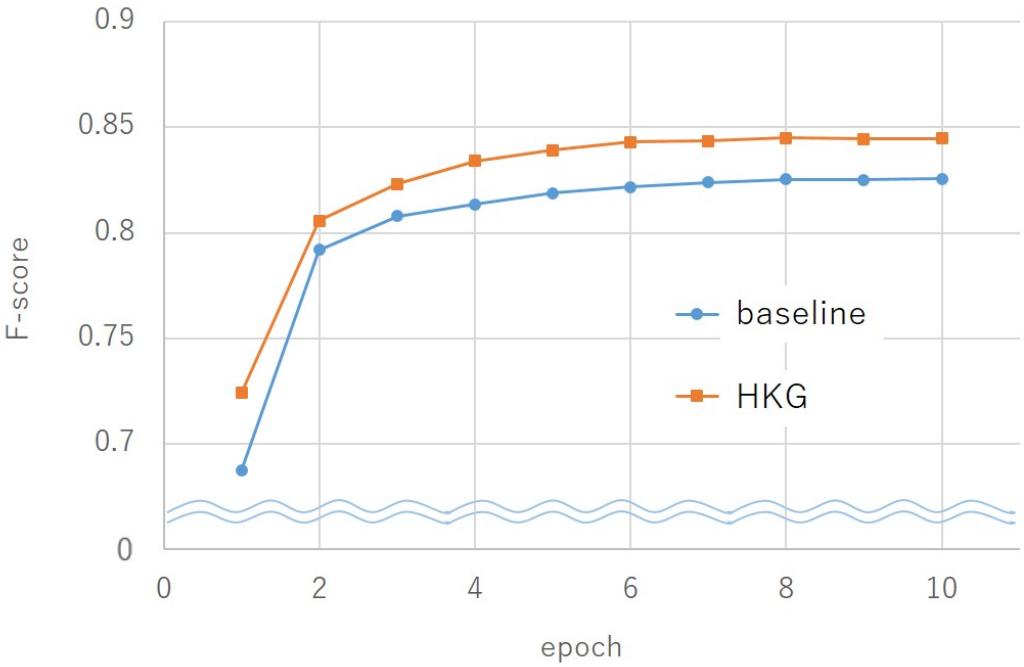


Figure 7.4: Learning curve of baseline model and the proposed model on 5-fold cross-validation data sets. The average of F-scores over the five validation data sets is reported.

model. From the 4th epoch, the model using KG embeddings outperforms the baseline in F-score more than 2 percent points.

7.4.6 Analysis of Prediction Results

The confusion matrices of the baseline and the proposed model are shown in Figure 7.5. The numbers indicate the total count of five validation data sets. The proposed method reduced the all patterns of errors (non-diagonal components in the table) compared to the baseline model. In particular, errors in which the model incorrectly predicts the *Effect* interaction as negative and errors in which the model incorrectly predicts the negative as *Effect* are greatly reduced. For *Mechanism*, *Advice*, and *Int.* relations, the use of heterogeneous KG information also reduced the number of cases of false negative or false positive relations. On the other hand, there were cases in which the use of heterogeneous KG information slightly increased errors by classifying relations into wrong types, e.g., incorrectly classifying a *Mechanism* relation as *Effect*.

In addition, four examples of prediction results are shown. Example 1, 2, and 3

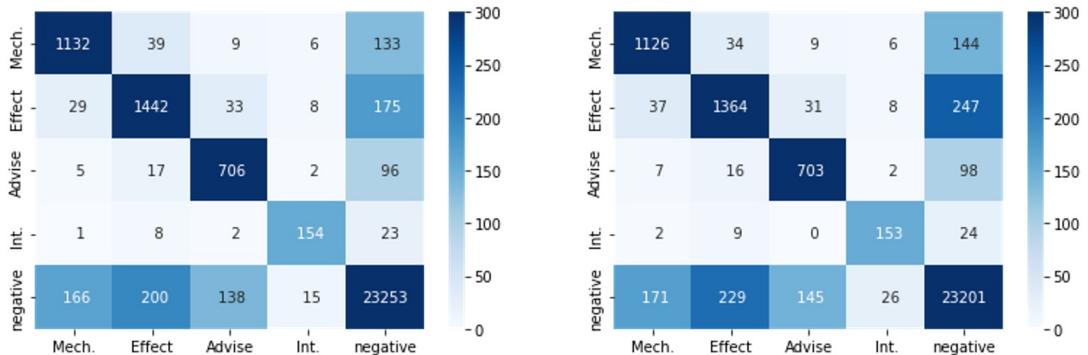


Figure 7.5: The confusion matrices (left: baseline, right: proposed method). The vertical axis shows the actual labels and the horizontal axis shows the predicted labels. The numbers indicates the total count of five validation data sets.

in Table 7.9 are cases correctly predicted by using heterogeneous KG information but incorrectly predicted by the baseline model. In Examples 1 and 2, many drug entities appear, and DRUG1 and DRUG2 are included in parentheses. As shown in Example 3, the baseline model tends to predict the cases where the distance between DRUG1 and DRUG2 is extremely short as negative, but the proposed model correctly predicts them. From these examples, heterogeneous KG representation of the drug entities may be helpful to predict correct relations when the prediction is difficult only from their surrounding contexts in the sentences.

Example 4 is a case incorrectly predicted by using heterogeneous KG information while correctly predicted by the baseline model. According to the annotation guideline of DDIExtraction-2013 data set, an interaction should only be annotated when it occurs in the text. Example 4 shows some studies about given interactions were performed, however, the sentence does not provide any evidence. In such a case, background knowledge of drugs may have disturbed correct prediction.

7.5 Summary

This chapter first added the molecular structure information of drugs to the KG data set constructed in the previous chapter. The results on the link prediction task showed that the MRR and Hits@k was improved by considering the molecular structure information of drugs.

Example 1

Text: In patients receiving nonselective DRUGOTHER (DRUGOTHER) (e.g., **DRUG1**) in combination with DRUGOTHER (e.g., DRUGOTHER, DRUGOTHER, DRUGOTHER, DRUGOTHER, **DRUG2**), there have been reports of serious, sometimes fatal, reactions.

DRUG1: selegiline hydrochloride, DRUG2: venlafaxine

Gold label: Effect **Baseline:** negative **Proposed model** Effect:

Example 2

Text: DRUGOTHER: In a study of 7 healthy male volunteers, **DRUG1** treatment potentiated the blood glucose lowering effect of DRUGOTHER (a DRUGOTHER similar to **DRUG2**) in 3 of the 7 subjects.

DRUG1: acitretin, DRUG2: chlorpropamide

Gold label: negative **Baseline:** Effect **Proposed model** negative:

Example 3

Text: Caution should be exercised when considering the use of **DRUG1** and **DRUG2** in patients with depressed myocardial function.

DRUG1: BREVIBLOC, DRUG2: verapamil

Gold label: Advice **Baseline:** negative **Proposed model** Advice:

Example 4

Text: To determine whether **DRUG1** has a direct effect on the distribution of **DRUG2**, the elimination and distribution of DRUGOTHER was studied in six patients, five lacking kidney function and one with a partially impaired renal function, in the presence or absence of DRUGOTHER.

DRUG1: probenecid, DRUG2: cloxacillin

Gold label: negative **Baseline:** negative **Proposed model** Mechanism:

Table 7.9: Case studies of the proposed model

Then, a method to use the heterogeneous KG representations for the relation extraction task was proposed. The proposed model incorporates heterogeneous KG embeddings into the input sentence in the form of levitated markers and considers the relationship between contexts and KG information through an attention mechanism. In the experiment, a significant improvement of 1.70 percent points on the DDIExtraction-2013 data set was achieved by using heterogeneous KG information.

8 Conclusions

8.1 Summary

This thesis proposed a novel approach that focuses on heterogeneous domain information for relation extraction from the literature. In case studies, the proposed method can perform DDI extraction integrally considering heterogeneous information about drugs, and showed the usefulness of utilizing various information about drugs for the DDI extraction task. After describing an attention mechanism for relation extraction in Chapter 3 and reviewing an approach to using a single kind of domain information in Chapter 4, this thesis devised neural relation extraction models that can consider heterogeneous domain information.

Chapter 5 proposed a novel neural method for relation extraction from text using large-scale raw text information and two kinds of domain information, which are 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 the literature on the DDIEtraction-2013 data set. 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.

Chapter 6 constructed a new heterogeneous knowledge graph containing textual information PharmaHKG from several databases. The combinations of three methods to use textual information and four scoring functions on the link prediction task are compared. The utility of text information and the best combination for the link prediction depends on the target relation types were found. In addition, when the averaged MRR for all relation types was in focus, a method that combines SimplE and text information achieved the highest MRR, and this result showed the usefulness of text information in the link prediction task in the drug domain.

Chapter 7 proposes a neural architecture that integrates sentence information and knowledge graph information constructed in Chapter 6. The molecular structure information of drugs is added to the KG data set constructed in the previous chapter. The results on the link prediction task showed that the MRR and Hits@k were improved by considering the molecular structure information of drugs. Then, a method to use the heterogeneous KG representations for the relation extraction task was proposed. The proposed model incorporates heterogeneous KG embeddings into the input sentence in the form of levitated markers and considers the relationship between contexts and KG information. In the experiment, an improvement of 1.70 percent points on the DDIExtraction-2013 data set was achieved by adding heterogeneous KG information.

8.2 Future Work

8.2.1 Employing the Deep Neural Entity Linking Method

As mentioned in Chapter 7, the coverage of linking drug mentions in the input sentence to drug entries in the KG is only about 90%, which has become a bottleneck in the overall DDI extraction model. In recent years, many deep neural entity linking models [109] have been proposed, and these models can classify the input mentions into KG entries even if the number of KG entries is more than several hundred thousand. The deep neural linking model has been reported to perform better than the model with simple string matching, and higher coverage and more accurate linking are expected by employing the deep neural linking model in the model.

In addition, when the issues about memory usage and training time are solved, a method that jointly train entity linking and DDI extraction can be realized. These ideas are left for future work.

8.2.2 Joint Learning of BERT and KG Embeddings

In the proposed method, the learning of KG embeddings is decoupled from the learning of DDI extraction. If the KG embeddings and BERT model are jointly updated, it

is expected that DDI label information can be taken into account when updating KG embeddings and the learning KG representation can be more suitable for DDI extraction.

Many GNN-based text classification methods [110, 111] have been proposed that perform the sentence classification task by placing words in the input sentence on a graph space and learning the graph representation. A possible extension of these models is to place the input sentences in a heterogeneous KG and train the word/sentence entities and other heterogeneous entities jointly to perform DDI extraction.

8.2.3 End-to-End DDI Extraction

DDI extraction from the literature consists of two parts: drug mention recognition and DDI extraction between the mention pair, and in this thesis, the author focuses on the DDI extraction part. Efficient simultaneous training of the two tasks may improve the performance of both tasks. It is important to analyze the impact of using heterogeneous information in the DDI extraction part on the mention recognition part.

Furthermore, drug mention recognition is also closely correlated with drug mention linking to KG entries. KG entries information can help to improve the drug mention recognition performance. The construction of a framework of combining the drug mention recognition part, drug mention linking part and DDI extraction part is left for future work.

8.2.4 Extension to Other Tasks

This thesis limits the experiment to DDI extraction and drug-protein interaction extraction. The main idea of the proposed model consists of two parts: representation of heterogeneous items in the form of KG and integration of representation embeddings by an attention mechanism. The proposed method can be extended to other tasks, since the method is not domain specific and domain information from many other fields can be applied to the input format of the model. The author would like to apply the proposed model to other tasks and test the versatility of the method.

Publications

- Using drug descriptions and molecular structures for drug-drug interaction extraction from literature [9]

Masaki Asada, Makoto Miwa, and Yutaka Sasaki, Bioinformatics volume 37, issue 12, pages 1739-1746, 2021.

- Representing a heterogeneous pharmaceutical knowledge-graph with textual information [10]

Masaki Asada, Nallappan Gunasekaran, Makoto Miwa, and Yutaka Sasaki, Frontiers in Research Metrics and Analytics, volume 6, pages 1-13, 2021.

- TTI-COIN at BioCreative VII Track 1 - Drug-protein interaction extraction with external database information [52]

Naoki Iinuma, Masaki Asada, Makoto Miwa, and Yutaka Sasaki, In Proceedings of BioCreative VII workshop, volume 1, pages 49-53, 2021.

- Integrating heterogeneous knowledge graphs into drug-drug interaction extraction from the literature [11]

Masaki Asada, Makoto Miwa, and Yutaka Sasaki, Bioinformatics, 2022.

References

- [1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.
- [2] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS 2013*, pages 3111–3119, 2013.
- [3] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186, Minneapolis, Minnesota, June 2019.
- [5] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *Proceedings of ICML 2011*, page 689–696, Madison, WI, USA, 2011. Omnipress.
- [6] Jing Zhao, Xijiong Xie, Xin Xu, and Shiliang Sun. Multi-view learning overview. *Inf. Fusion*, 38(C):43–54, nov 2017.
- [7] Council on Family Health. *DRUG INTERACTIONS: WHAT YOU SHOULD KNOW*. Federal Citizen Information Center, 2001.
- [8] David L Sackett. Evidence-based medicine. In *Seminars in perinatology*, volume 21, pages 3–5. Elsevier, 1997.
- [9] Masaki Asada, Makoto Miwa, and Yutaka Sasaki. Using drug descriptions and molecular structures for drug–drug interaction extraction from literature. *Bioinformatics*, 37(12):1739–1746, 10 2020.

- [10] Masaki Asada, Nallappan Gunasekaran, Makoto Miwa, and Yutaka Sasaki. Representing a heterogeneous pharmaceutical knowledge-graph with textual information. *Front. Res. Metr. Anal.*, 6:670206, July 2021.
- [11] Masaki Asada, Makoto Miwa, and Yutaka Sasaki. Integrating heterogeneous knowledge graphs into drug-drug interaction extraction from the literature. *Bioinformatics*, 2022.
- [12] Sachin Pawar, Girish K. Palshikar, and Pushpak Bhattacharyya. Relation extraction : A survey. *CoRR*, abs/1712.05191, 2017.
- [13] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon, 2004.
- [14] Xavier Carreras and Lluís Màrquez. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of CoNLL 2004*, pages 89–97, Boston, Massachusetts, USA, May 6 - May 7 2004.
- [15] Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden, July 2010.
- [16] Razvan Bunescu, Ruifang Ge, Rohit J Kate, Edward M Marcotte, Raymond J Mooney, Arun K Ramani, and Yuk Wah Wong. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial intelligence in medicine*, 33(2):139–155, 2005.

- [17] Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. Comparative analysis of five protein-protein interaction corpora. In *BMC bioinformatics*, volume 9, pages 1–11. BioMed Central, 2008.
- [18] Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Proceedings of SemEval 2013*, pages 341–350, Atlanta, Georgia, USA, June 2013.
- [19] Martin Krallinger, Obdulia Rabal, Saber Ahmad Akhondi, Martín Pérez Pérez, Jesus Santamaría, Gael Pérez Rodríguez, Georgios Tsatsaronis, Ander Intxaurrondo, J. A. Lopez, Umesh K. Nandal, Erin M. van Buel, Ambika Chandrasekhar, Marleen Rodenburg, Astrid Lægreid, Marius A. Doornenbal, Julen Oyarzábal, Anália Lourenço, and Alfonso Valencia. Overview of the biocreative vi chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, pages 141–146, 2017.
- [20] Antonio Miranda, Farrokh Mehryary, Jouni Luoma, Sampo Pyysalo, Alfonso Valencia, and Martin Krallinger. 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.
- [21] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014*, pages 2335–2344, Dublin, Ireland, August 2014.
- [22] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

- [23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Com-
put.*, 9(8):1735–1780, nov 1997.
- [24] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8*, pages 103–111, Doha, Qatar, October 2014.
- [25] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP 2012*, pages 1201–1211, Jeju Island, Korea, July 2012.
- [26] Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. Drug-drug interaction extraction via convolutional neural networks. *Computational and mathematical methods in medicine*, 2016.
- [27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of NeurIPS 2017*, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [28] Masaki Asada, Makoto Miwa, and Yutaka Sasaki. Extracting drug-drug interactions with attention CNNs. In *BioNLP 2017*, pages 9–18, Vancouver, Canada,, August 2017.
- [29] Md Faisal Mahbub Chowdhury and Alberto Lavelli. FBK-irst: A multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information. *Atlanta, Georgia, USA*, 351:53, 2013.
- [30] Zhehuan Zhao, Zhihao Yang, Ling Luo, Hongfei Lin, and Jian Wang. Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. *Bioinformatics*, 32(22):3444–3453, 2016.

- [31] Sunil Kumar Sahu and Ashish Anand. Drug-drug interaction extraction from biomedical texts using long short-term memory network. *Journal of biomedical informatics*, 86:15–24, 2018.
- [32] Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation classification via multi-level attention CNNs. In *Proceedings of ACL 2016*, pages 1298–1307, 2016.
- [33] Sangrak Lim, Kyubum Lee, and Jaewoo Kang. Drug drug interaction extraction from the literature using a recursive neural network. *PloS one*, 13(1):e0190926, 2018.
- [34] Cícero dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. In *Proceedings of ACL-IJCNLP 2015*, pages 626–634, Beijing, China, July 2015.
- [35] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of ICML 2010*, pages 111–118, 2010.
- [36] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- [37] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of ICLR 2015*, 2015.
- [38] Isabel Segura-Bedmar, Paloma Martínez Fernández, and Daniel Sánchez Cisneros. The 1st ddiextraction-2011 challenge task: Extraction of drug-drug interactions from biomedical texts. In *CEUR Workshop Proceedings*, volume 461, pages 1–9. CEUR-WS.org, 2011.
- [39] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, Nazanin Assempour, Ithayavani Iynkkaran, Yifeng Liu, Adam Maciejewski, Nicola Gale, Alex Wilson,

Lucy Chin, Ryan Cummings, Diana Le, Allison Pon, Craig Knox, and Michael Wilson. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic Acids Research*, 46(D1):D1074–D1082, 11 2017.

- [40] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP 2014*, pages 1746–1751, Doha, Qatar, October 2014.
- [41] Chanqin Quan, Lei Hua, Xiao Sun, and Wenjun Bai. Multichannel convolutional neural network for biological relation extraction. *BioMed Research International*, 2016:1850404, Dec 2016.
- [42] Sun Kim, Haibin Liu, Lana Yeganova, and W John Wilbur. Extracting drug-drug interactions from literature using a rich feature-based linear kernel approach. *Journal of biomedical informatics*, 55:23–30, 2015.
- [43] Masaki Asada, Makoto Miwa, and Yutaka Sasaki. Enhancing drug-drug interaction extraction from texts by molecular structure information. In *Proceedings of ACL 2018*, pages 680–685, Melbourne, Australia, July 2018.
- [44] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional Networks on Graphs for Learning Molecular Fingerprints. In *Proceedings of NeurIPS 2015*, pages 2224–2232, 2015.
- [45] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *Proceedings of ICLR 2016*, 2016.
- [46] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *Proceedings of ICML 2017*, pages 1263–1272, 2017.

- [47] Yoshimasa Tsuruoka, Yuka Tateishi, Jin-Dong Kim, Tomoko Ohta, John McNaught, Sophia Ananiadou, and Jun’ichi Tsujii. Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic Conference on Informatics*, pages 382–392, 2005.
- [48] David Weininger. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.
- [49] Greg Landrum. RDKit: Open-source cheminformatics, 2016.
- [50] Wei Zheng, Hongfei Lin, Ling Luo, Zhehuan Zhao, Zhengguang Li, Yijia Zhang, Zhihao Yang, and Jian Wang. An attention-based effective neural model for drug-drug interactions extraction. *BMC Bioinformatics*, 18(1):445, Oct 2017.
- [51] Ronald Aylmer Fisher et al. The design of experiments. *The design of experiments.*, 1937.
- [52] Iinuma Naoki, Masaki Asada, Makoto Miwa, and Yutaka Sasaki. TTI-COIN at BioCreativeVII Track 1. In *Proceedings of sixth BioCreative Challenge Evaluation Workshop*, pages 49–53, Online, November 2021.
- [53] Masashi Tsubaki, Kentaro Tomii, and Jun Sese. Compound–protein interaction prediction with end-to-end learning of neural networks for graphs and sequences. *Bioinformatics*, 35(2):309–318, 2019.
- [54] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of EMNLP-IJCNLP 2019*, pages 3615–3620, Hong Kong, China, November 2019.
- [55] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of EMNLP 2018*, pages 66–71, Brussels, Belgium, November 2018.

- [56] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *CoRR*, abs/1606.08415, 2016.
- [57] Greg Landrum. RDKit: Open-source cheminformatics software, 2016.
- [58] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of ICLR 2019*, 2019.
- [59] Manuel Le Gallo, Abu Sebastian, Roland Mathis, Matteo Manica, Heiner Giefers, Tomas Tuma, Costas Bekas, Alessandro Curioni, and Evangelos Eleftheriou. Mixed-precision in-memory computing. *Nature Electronics*, 1(4):246–253, 2018.
- [60] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.
- [61] Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of BioNLP 2019*, pages 58–65, Florence, Italy, August 2019.
- [62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [63] Wei Wang, Xi Yang, Canqun Yang, Xiaowei Guo, Xiang Zhang, and Chengkun Wu. Dependency-based long short term memory network for drug-drug interaction extraction. *BMC Bioinformatics*, 18(16):578, Dec 2017.
- [64] Naoki Iinuma, Makoto Miwa, and Yutaka Sasaki. Improving supervised drug-protein relation extraction with distantly supervised models. In *Proceedings of BioNLP 2022*, pages 161–170, Dublin, Ireland, May 2022.

- [65] The UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, 11 2018.
- [66] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado, June 2015.
- [67] Jinyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 09 2019.
- [68] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of ICML 2011*, page 809–816, Madison, WI, USA, 2011. Omnipress.
- [69] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of AAAI 2011*, page 301–306. AAAI Press, 2011.
- [70] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. *Advances in Neural Information Processing Systems*, 26:926–934, 2013.
- [71] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of AAAI 2016*, volume 30, 2016.
- [72] Duc-Hong Pham and Anh-Cuong Le. Learning multiple layers of knowledge representation for aspect based sentiment analysis. *Data & Knowledge Engineering*, 114:26–39, 2018.
- [73] Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Representation learning of knowledge graphs with hierarchical types. In *IJCAI*, pages 2965–2971, 2016.

- [74] Yankai Lin, Zhiyuan Liu, and Maosong Sun. Knowledge representation learning with entities, attributes and relations. *ethnicity*, 1:41–52, 2016.
- [75] Zizheng Ji, Zhengchao Lei, Tingting Shen, and Jing Zhang. Joint representations of knowledge graphs and textual information via reference sentences. *IEICE Transactions on Information and Systems*, 103(6):1362–1370, 2020.
- [76] Ni Lao, Tom Mitchell, and William Cohen. Random walk inference and learning in a large scale knowledge base. In *Proceedings of EMNLP 2011*, pages 529–539, 2011.
- [77] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI 2015*, page 2181–2187. AAAI Press, 2015.
- [78] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Ok-sana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of NuerIPS 2013*, pages 1–9, 2013.
- [79] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.
- [80] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, pages 2022–2032, 2019.
- [81] Xu Han, Zhiyuan Liu, and Maosong Sun. Joint representation learning of text and knowledge for knowledge graph completion. *arXiv preprint arXiv:1611.04125*, 2016.
- [82] Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *Proceedings of EMNLP 2015*, pages 1499–1509, 2015.

- [83] Yashen Wang, Huanhuan Zhang, Ge Shi, Zhirun Liu, and Qiang Zhou. A model of text-enhanced knowledge graph representation learning with mutual attention. *IEEE Access*, 8:52895–52905, 2020.
- [84] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.
- [85] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of WWW 2017*, pages 697–706, 2007.
- [86] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [87] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of ICLR 2014*, 2014.
- [88] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR 2017*. OpenReview.net, 2017.
- [89] Timothy Jewison, Yilu Su, Fatemeh Miri Disfany, Yongjie Liang, Craig Knox, Adam Maciejewski, Jenna Poelzer, Jessica Huynh, You Zhou, David Arndt, Yanick Djoumbou, Yifeng Liu, Lu Deng, An Chi Guo, Beomsoo Han, Allison Pon, Michael Wilson, Shahrzad Rafatnia, Philip Liu, and David S Wishart. SMPDB 2.0: big improvements to the Small Molecule Pathway Database. *Nucleic Acids Res*, 42(Database issue):D478–484, Jan 2014.
- [90] C. E. Lipscomb. Medical Subject Headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265–266, Jul 2000.

- [91] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, pages 2071–2080. PMLR, 2016.
- [92] Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. In *Proceedings of NeurIPS 2018*, pages 4289–4300, 2018.
- [93] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.
- [94] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [95] Da Zheng, Xiang Song, Chao Ma, Zeyuan Tan, Zihao Ye, Jin Dong, Hao Xiong, Zheng Zhang, and George Karypis. DGL-KE: Training knowledge graph embeddings at scale. In *Proceedings of SIGIR 2020*, page 739–748, New York, NY, USA, 2020.
- [96] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrick Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and

- Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP 2020*, pages 38–45, Online, October 2020.
- [97] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of ACL 2019*, pages 1441–1451, Florence, Italy, July 2019.
- [98] Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *Proceedings of EMNLP-IJCNLP 2019*, pages 43–54, Hong Kong, China, November 2019.
- [99] Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. In *Proceedings of NAACL 2021*, pages 50–61, Online, June 2021.
- [100] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of ACL 2019*, pages 2895–2905, Florence, Italy, July 2019.
- [101] Chaojun Xiao, Yuan Yao, Ruobing Xie, Xu Han, Zhiyuan Liu, Maosong Sun, Fen Lin, and Leyu Lin. Denoising relation extraction from document-level distant supervision. In *Proceedings of EMNLP 2020*, pages 3683–3688, Online, November 2020.
- [102] Seyone Chithrananda et al. Chemberta: Large-scale self-supervised pretraining for molecular property prediction. In *Machine Learning for Molecules Workshop at NeurIPS 2020*, 2020.
- [103] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12(null):2121–2159, jul 2011.

- [104] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Research*, 49(D1):D1388–D1395, 11 2020.
- [105] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.
- [106] David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.
- [107] Boris T Polyak and Anatoli Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.
- [108] Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. Scifive: a text-to-text transformer model for biomedical literature. *CoRR*, abs/2106.03598, 2021.
- [109] Dongfang Xu and Timothy Miller. A simple neural vector space model for medical concept normalization using concept embeddings. *Journal of Biomedical Informatics*, 130:104080, 2022.
- [110] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of AAAI'19/IAAI'19/EAAI'19*. AAAI Press, 2019.
- [111] Yufeng Zhang, Xueli Yu, Zeyu Cui, Shu Wu, Zhongzhen Wen, and Liang Wang. Every document owns its structure: Inductive text classification via graph neural networks. In *Proceedings of ACL 2020*, pages 334–339, Online, July 2020.