creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

M.S. THESIS

Graph Neural Network based Knowledge Graph Completion for Predicting Drug-Drug Interaction

그래프 신경망 기반 지식 그래프 완성을 통한 약물 상호작용 예측

BY

Park Sangha

February 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

M.S. THESIS

Graph Neural Network based Knowledge Graph Completion for Predicting Drug-Drug Interaction

그래프 신경망 기반 지식 그래프 완성을 통한 약물 상호작용 예측

BY

Park Sangha

February 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Graph Neural Network based Knowledge Graph Completion for Predicting Drug-Drug Interaction

그래프 신경망 기반 지식 그래프 완성을 통한 약물 상호작용 예측

지도교수 문 봉 기

이 논문을 공학석사 학위논문으로 제출함

2021년 2월

서울대학교 대학원

컴퓨터 공학부

박 상 하

박상하의 공학석사 학위 논문을 인준함

2021년 2월

위 원 장: 김 형 주 (인)

부위원장: 문 봉 기 (인)

위 원: 이 상 구 (인)

Abstract

Previous studies reported various computational drug representation methods for predicting drug-drug interactions to avoid side effects induced by taking multiple drugs together. Proteins as targets, enzymes, transporters and carriers cause interactions and thus are used as a feature for the drug representation. However, previous drug representation methods do not extract enough information to predict drug interactions and are limited only to detect interactions between two drugs but not a quantification of interactions. This paper presents a novel Drug Graph Completion (DGC) for (1) an improved drug representation and (2) a prediction of quantifying drug interactions. DGC is the model well-suitable for predicting an increase or decrease (quantification) of drug interactions by reflecting drug-protein relations. This model consists of Graph Neural Network (GNN) and Knowledge Graph Completion (KGC) act as encoder-decoder, respectively. First, Graph Attention Network, one of GNN, generates drug vectors by assigning different importance between neighbor nodes such that a node affecting interactions receives a higher attention value. Second, the Knowledge Graph Completion (KGC) method, one of the link prediction models, is applied to quantify drug interactions increasing or decreasing. KGC predicts by calculating the validity of triple, consisting of two drug vectors and a vector representing an amount of their interaction. Experimental results demonstrate significant predictive accuracy improvement compared to previous drug-drug interaction prediction methods and the KGC and GNN model. In addition, the validation results show that our model successfully predicts the quantification of drug interactions.

keywords: Graph Neural Network, Knowledge Graph Completion, Drug Drug Interaction

student number: 2019-22256

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Drug-drug interaction (DDI) refers to the unexpected symptoms, or side effects, of taking two different drugs together. Drug interactions are divided into two groups, pharmacokinetic (PK) and pharmacodynamic (PD), and we mainly focused on PK interactions in this study. PK is the body's response to a drug, which includes absorption, distribution, metabolism, and excretion (ADME). Cytochrome P450 (CYP450) enzymes are essential for the metabolism of many drugs, and they can be inhibited, induced or substrated by drugs. Co-administering two drugs (one for an inhibitor and another for a substrate of the enzyme) result in clinically significant drug-drug interactions that can cause unanticipated adverse reactions or therapeutic failures. Drug interactions can be inferred by a change in the concentration of the drugs. In other words, the amount of variation in the serum concentration detects interactions. In general, inhibition of drug metabolism elevates concentrations whereas induction decreases the concentration.

The increasing number of approved drugs has made drug interactions more likely, especially for patients taking multiple drugs, such as cancer patients. The unexpected side effects caused by DDIs are hazardous (may lead to deaths) and significantly increase healthcare costs. Therefore, DDI studies have always attracted much attention to drug safety and healthcare management [1]. Existing DDI studies have focused on

metabolic profile tests such as CYP450 or transporter-associated pharmacokinetic interaction [2]. However, a limitation of DDI research based on experimental approach such as insufficient experimental data, high cost of research, long time required, consideration of animal protection, etc., is a major obstacle in the drug development phase [3]. In that sense, various computational methods has been proposed than simulation methods. Simulation methods represented by PBPK study required mathematical equations to describe ADME's properties resulting in numerous parameters and thus longer time for the analyses. [4, 5].

Several computational methods such as *similarity-based*, *network-based*, *matrix factorization-based*, and *Graph Neural Network-based* approaches were proposed to predict DDI. First, the *similarity-based* approach [6, 2, 7, 8, 9] assumes that two similar drugs interact. Second, the *network-based* approach [10, 11] calculates similarities such as common neighbor, Adamic-Adar, Resource Allocation and Katz similarity in the network adjacency matrix. Third, the *matrix factorization-based* approach [12, 13, 14] decompose the adjacency matrix and reconstruct the adjacency matrix to identify novel DDIs. Finally, there are several methods using *Graph Neural Networks (GNN)*. [15] use Graph Convolutional Network to represent molecular graph structures. [16] model each drug as a node in the drug association network and extend the Graph Convolutional Networks (GCN) to embed features such as DDIs, side effects and chemical structures. Recent studies also adopted Knowledge Graph (KG) for a DDI prediction. [17] proposes a method to obtain the rich neighborhood information of each entity in KG by learning from the neighborhoods for each entity as their local receptive and integrating neighborhood information with bias from representing the current entity. However, these aforementioned approaches accomplished a limited success as they could only predict an existence of interactions while the amount of interactions was not quantified.

Recently, new methods are proposed to predict an interaction more specifically. [18] provide fine-grained descriptions including drug-drug interaction mechanism and

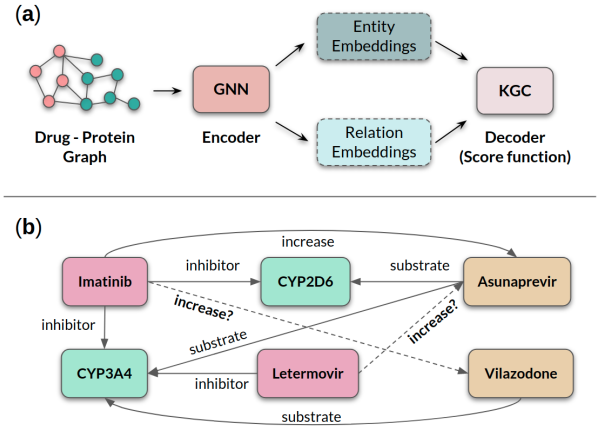

Figure 1.1: (a) Link prediction process; the encoder generates both entity and relation embeddings and decoder returns score to determine whether the triple is valid. (b) Subgraph of a Knowledge Graph contains actual relations between entities (solid lines) and inferred relations that are initially hidden (dashed lines) (example of increase case). A triple example of this subgraph is as follows: (Imatinib, inhibitor, CYP2D6)

action using pathway, substructure, target and enzyme. Also, [19] presents complex relationships of drug interactions by providing side effects. These methods still have some limitations. Although [18] predicts various DDIs, their prediction is not based on the underlying mechanism of DDIs and not focused on the amount of serum concentration increasing or decreasing. Both of them use proteins as features; they only consider drug-protein relation as a binary outcome - exist or not.

With all these limitations noted, in this study, we propose and demonstrate a GNN based Knowledge Graph Completion (KGC) model, DGC (Drug Graph Completion), the model for predicting an increase/decrease of DDIs based on drug-protein relations. Estimating an amount of DDIs (increase/decrease) is essential to optimizing patient care, setting up drug doses and finding drug resistance in a polypharmacy environment. We construct a graph with drugs, proteins and their interactions and then embed nodes and relations using the Graph Attention Network (GAT), a GNN model. Then KGC

calculates a score of how valid the triple is to construct a link which represents DDI. The main advantage of this method is that DGC not only predicts an existence of DDIs but also can quantify the interactions. DGC utilizes multi relations between drugs and proteins such as inhibit and induce, which incur DDIs to quantify interactions. In addition, this model shows high accuracy in predicting DDIs compared to the other previous methods.

Main contributions of this study are as followings:

- We propose an informative drug embedding method that reflects inhibit/induce/substrate information closely related to DDI. Furthermore, our model predicts not only whether interaction occurs or not but also the amount of the interaction (quantification).
- Our proposed DGC model, Graph Attention Network-based KGC method, shows the best performance. We demonstrate the performance of our model by comparing with other previous DDI methods which only employed either KGC or GNN model.
- It is validated that our model is able to quantify the DDIs by comparing the interactions predicted by the model and previously reported.

Chapter 2

Materials

2.1 Overview

Fig. 1.1a shows the link prediction (DDI prediction) process of DGC. First, we construct a Drug-Protein graph, which is a Knowledge Graph. A Knowledge Graph (KG) is a graph consisting of entity and relation, expressed in triple form *(head, relation, tail)* (Fig. 1.1b). This triple means entity *head* has a relationship *relation* with entity *tail*. Then, the encoder (GNN) embeds entities and relations of KG. The embedding method is the process of learning to express the relations and the entities effectively. Lastly, with the encoder's output, the decoder (KGC) calculates the score with the score function. The scoring function f calculates how valid triple is when the triple enters the input value.

2.2 Graph Construction

We start by explaining the Graph Construction process. At first, we describe the most critical information, drug-CYP information, and then analyze supporting information to predict the quantification of drug interaction.

2.2.1 Drug-CYP Interaction

There are three known types of interactions between drugs and CYPs: the drug is

- a substrates, metabolized by the CYP enzyme, or
- an inhibitor, inhibits the CYP activity, or
- an inducer, increases the CYP activity,

and some drugs interact with a CYP in more than one way. Predicting drug interaction and quantification of it is closely related to drug-CYP relation. If a drug pair interacts, the subject of the interaction drug is a perpetrator, and the other affected drug is a victim. The perpetrator drug inhibits or induces CYP while the affected drug substrates CYP. As mentioned before, in the case of a perpetrator is an inhibitor, the concentration of drug value is increased. On the other hand, if a perpetrator is an inducer, the concentration value is decreased. For this reason, the *inhibit/induce/substrate* of drug metabolism enzymes is an important drug-drug interaction source. CYP3A4, as an example of enzymes, is involved in the metabolism of numerous drugs, and CYP induction is a major concern in clinical practice.

2.2.2 Supporting Information

Predicting increase or decrease of interaction is challenging due to several reasons. Although we decide to use drug-CYP interaction to predict the quantification of drug interactions, not every increasing/decreasing drug interaction is involved with drug metabolism enzymes. Only 44% of known interactions are related to these relations. Furthermore, sometimes drug interaction can happen paradoxically. If a drug's metabolism gives rise to a product, which produces the effect of the drug, the enzymatic inhibition causes a decrease in the drug's effect. Therefore, to predict the quantification of drug interactions, we need supporting information such as other drug and protein relations, not only drug enzyme metabolism, or drug and drug relations, protein and protein relation. We looked at how each relation plays a role in predicting drug interactions.

- **Drug-Protein Interactions**: Co-prescribed drugs tend to have more proteins in common than random pairs. [19] This fact suggests drug-protein interaction information is valuable for predicting drug interactions. The inhibition of metabolizing enzymes produces many interactions, but other possible mechanisms are also produced, such as interactions of transporters or pharmacological targets. Therefore we include other proteins (target, transporter, and carrier) and other drug-protein relations (*antagonist* and *agonist*) in our graph. An *agonist* is a chemical that binds to a receptor and activates the receptor to produce a biological response while an *antagonist* blocks the agonist's action. We also use *others* (ex. antibody, activator, or modulator) to include drug-protein relationship not belonging to any of the above relations. Including *inhibit/induce/substrate*, in DGC graph, the number of drug-protein relations is six.

- **Drug-Drug Interactions**: Some previous studies show that known drug interaction information can be used as sufficient information to predict new interactions. For example, by including interaction profile fingerprint-based similarity, [20] constructed a large-scale drug interaction predictor. Vilar's model considers different pharmacological effects implicated in the drug interaction information. By proposing a new interaction prediction model using only information about the interacting drugs, [20] shows drug-drug interaction incorporates implicit bioavailability information. For this reason, we include drug-drug interaction to predict unknown DDIs. There are two relations between drugs, *increase* and *decrease*.

- **Protein-Protein Interactions**: As we mentioned before, most co-prescribed drugs have common proteins. However, there are more than 11% of drug combinations with zero target proteins in common. In this case, drugs cannot connect each other with the drug-protein relationship. This fact suggests that it is important to use protein-protein interaction information to connect different proteins

targeted by various drugs. According to [19], considering how proteins interact with each other and modeling longer chains of (indirect) interactions are essential to predict novel drug interactions. Therefore, we include protein-protein interaction information to illustrate drug interaction mechanisms better and predict DDIs. We refer to cases such that protein-protein interaction makes a longer chain of interacting two drugs in Section 3 of Appendix. This interaction is defined in one form, and we named it *PPI*.

The final DGC drug-protein graph consists of multi-type entities and relations. The elements of the entity set defined in this study are drugs and proteins, totaling two, and the relation set elements are six **drug and protein** interactions, two **drug and drug** interactions, and one **protein and protein** interaction, totaling nine. According to Table 2.1 notations, our graph G statistic is $|A| = 2$ and $|R| = 9$, and the graph include the node type set A = {*drug, protein*}, and the relation type set R = {*increase, decrease, antagonist, agonist, substrate, inhibit, induce, others, PPI*}.

Table 2.1: Graph Notations

Notations	Descriptions
$\lvert \cdot \rvert$	The cardinality of a set
$\mathrm{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$	Graph G with nodes set $\mathcal{V}$ and edges set $\mathcal{E}$
A	the set of node types ($\phi(v) = p \in A$)
R	the set of relation types
e = (*i, r, j*)	a edge e from vertex $i \in V$ to $j \in V$ with a relation type $r \in R$

Chapter 3

Methods

Using the constructed graph G, DGC predicts labeled edges between drug nodes. In other words, given a drug pair (v_i, v_j), i.e., $v_i, v_j \in \{drug\}$, our aim is to determine how likely a triple $t_{ij} = (v_i, r, v_j)$ of type r belongs to *increase, decrease*. DGC has two main components (Fig. 1.1a):

- Graph Neural Network (an encoder): a Graph Attention Network operating on G and producing embeddings for nodes in G and
- Knowledge Graph Completion (a decoder): a Convolutional Neural Network (CNN) model using these embeddings to model increase/decrease interactions.

We proceed by describing these two part, our approach for predicting drug-drug interactions with serum concentration changes using drug-protein graph.

3.1 Graph Attention Network

We first describe the graph encoder model, Graph Attention Network (GAT). The input to the GAT is node feature set $\mathbf{h} = \left\{\vec{h_1}, \vec{h_2}, ..., \vec{h_N}\right\}$, where $\vec{h_i} \in \mathbb{R}^F$, where F represents the feature dimension of each node and N represents the number of nodes. At least one learnable linear transformation is required to convert input features to

higher-level, where the weight matrix used is $\mathbf{W} \in \mathbb{R}^{F' \times F}$. The self-attention mechanism [21] for calculating the importance of node j for node i is calculated using $e_{ij} = a(\mathbf{W}\vec{h_i}, \mathbf{W}\vec{h_j}), j \in \mathcal{N}_i$. The obtained coefficient is normalized with softmax function:

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}. \tag{3.1}$$

The attention coefficient e_{ij} is $\mathrm{LeakyReLU}(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h_i}||\mathbf{W}\vec{h_j}])$, and $\mathbf{a}$ is a single layer feedforward network $\vec{\mathbf{a}} \in \mathbb{R}^{2F'}$. This value is used to calculate the output feature of a node:

$$\vec{h'_i} = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}\mathbf{W}\vec{h_j}\right). \tag{3.2}$$

One layer returns a new set of features $\mathbf{h'} = \left\{\vec{h'_1}, \vec{h'_2}, ..., \vec{h'_N}\right\}$, where $\vec{h'_i} \in \mathbb{R}^{F'}$, to output.

3.2 GAT with Relation

There are two main differences between the existing GAT and DGC's encoder. In DGC, our graph includes essential information in relations. Therefore, node embedding must represent relation information, and relation embedding itself is needed. We proceed by explaining each embedding process. Our model borrows the idea of a KBAT [22].

In each layer, two embedded matrices $\mathbf{h} = \left\{\vec{h_1}, \vec{h_2}, ..., \vec{h_{N_e}}\right\}$ for entity, where $\vec{h_i} \in \mathbb{R}^T$ and $\mathbf{g} = \{\vec{g_1}, \vec{g_2}, ..., \vec{g_{N_r}}\}$ for relation, where $\vec{g_i} \in \mathbb{R}^P$, are received as inputs. N_e and N_r are the number of entities and relations respectively, T and P are the dimensions of each. In DGC graph, N_e and N_r are the numbers of entities and relations mentioned in Section 4.1. To initialize embedding values, we use TransE [23] model and set the dimensions (T and P) as 50.

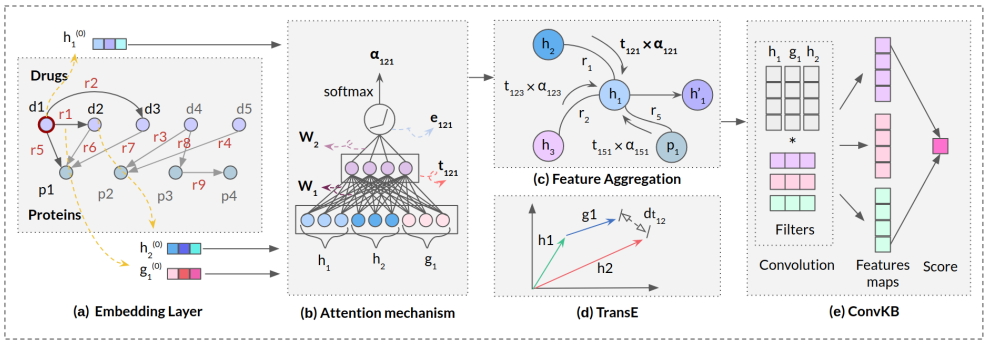

Figure 3.1: Overview of DGC model architecture. (a) DGC constructs a graph with two entities and nine relations. (b) The attention mechanism $a\mathbf{W}(\vec{h_i}, \vec{h_j}, \vec{g_k})$ parameterized by a weight matrix, applying a LeakyReLU activation. (c) To obtain the final output, aggregate all the neighbor triples of the node. (d) Translational scoring function to learn entity and relation embedding (h_2 needs to be the nearest neighbor of h_1 connected via relation g_1). (e) Process involved in ConvKB (with the embedding size k=4 and the number of filters Ω = 3 for illustration purpose).

3.2.1 Entity and Relation Embedding

For **entity embedding**, DGC's encoder calculates the attention coefficient for the neighbor *triple*, $e_{ijk} = a\mathbf{W}(\vec{h_i}, \vec{h_j}, \vec{g_k})$ while the existing GAT calculated the attention coefficient for the neighbor *node*. Therefore, embedded results can include relation information as well as a neighbor node. The detailed process to get attention coefficient is as follows: first, the neighbor triple $\vec{t_{ijk}}$ is obtained by concatenating two node embedding vectors and one relation embedding vector and multiply linear transformation $\mathbf{W}$, i.e., $\vec{t_{ijk}} = \mathbf{W_1}[\vec{h_i} \parallel \vec{h_j} \parallel \vec{g_k}]$. Then, this triple value becomes the input of the single layer feedforward network, $e_{ijk} = \text{LeakyReLU}\left(\mathbf{W_2}\vec{t_{ijk}}\right)$. The normalize process utilizes the softmax function just like the normal GAT (Fig. 3.1b),

$$\alpha_{ijk} = \frac{\exp\left(e_{ijk}\right)}{\sum_{n\in\mathcal{N}_i}\sum_{r\in\mathcal{R}_{in}}\exp\left(e_{inr}\right)}, \tag{3.3}$$

and the concatenate process for the output feature vector is applied to the neighbor *triples* instead of the neighbor *nodes* as follows (Fig. 3.1c):

$$\vec{h_i'} = \sigma\left(\sum_{j\in\mathcal{N}_i}\sum_{k\in\mathcal{R}_{ij}}\alpha_{ijk}\vec{t_{ijk}}\right). \tag{3.4}$$

For **relation embedding**, the relation vector is updated by $G' = G\cdot\mathbf{W}_R$, where $\mathbf{W}_R \in \mathbb{R}^{P\times P'}$. The result of one layer for entity and relation embedding vectors are $\mathbf{h'} = \left\{\vec{h_1'}, \vec{h_2'}, ..., \vec{h_{N_e}'}\right\}$ $(\vec{h_i'} \in \mathbb{R}^{T'})$ and $\mathbf{g'} = \left\{\vec{g_1'}, \vec{g_2'}, ..., \vec{g_{N_r}'}\right\}$ $(\vec{g_i'} \in \mathbb{R}^{P'})$ respectively.

3.2.2 Traning Encoder

For training learnable linear transformations, $\mathbf{W_1}, \mathbf{W_2}$ and $\mathbf{W_R}$, we adopt TransE (Fig. 3.1d) model. The idea of TransE [23] is when there is valid triple $t_{ij}^k = (h_i, g_k, h_j)$, they must satisfy $\vec{h_i} + \vec{g_k} \approx \vec{h_j}$. For $d_{t_{ij}^k}$ $(= \|\vec{h_i} + \vec{g_k} - \vec{h_j}\|_1)$, the smaller the value for a valid set and the larger for the invalid set, the more consistent the idea of TransE. Thus, the loss function is defined as Equation (3.5), and the difference between valid

triple and corrupted triple is further learned by having a margin:

$$\mathcal{L} = \sum_{t_{ij}^k \in S} \sum_{{t'_{ij}}^k \in S'} \max\left\{ d_{t_{ij}^k} - d_{{t'_{ij}}^k} + \gamma, 0 \right\} \tag{3.5}$$

$$\text{in which } S' = \{(h_{i'}, g_k, h_j) | h_{i'} \in V\} \cup \left\{(h_i, g_k, h_{j'}) | h_{j'} \in V\right\}.$$

The S is set for the valid triple on the DGC Graph, while S' is the corrupted triple set that dose not appear in the DGC Graph. The set of corrupted triples is composed of training triples with either the head or tail replaced by a random entity (but not both at the same time) [23].

3.3 Knowledge Graph Completion

Now we describe our decoder model, Knowledge Graph Completion (KGC). KGC, one of link prediction methods, deals with prediction of new facts (i.e., triples (*h, r, t*)). We'll explain the overall KGC methods first and then look at DGC's decoder model.

3.3.1 Background

The link prediction process by KGC can be explained as follows. Formally, the Knowledge Graph is represented by a directed, labeled graph G = $(\mathcal{V}, \mathcal{E}, \mathcal{R})$. By assuming that we are given only an incomplete subset $\hat{\mathcal{E}}$ rather than the full set of edges $\mathcal{E}$, we conduct the link prediction task. The task is to assign scores $f(h, r, t)$ to possible edges (or triples) (h, r, t) in order to determine how likely those edges are to belong to $\mathcal{E}$ [24].

KGC methods are largely divided into the *Factorization models* [25, 26, 27], *Translational models* [23, 28, 29], and *Neural network models* [30, 31]. First, *Translational models* formulate relations as a linear/bilinear mapping by projecting head entities into a representation space close to tail entities. Second, *Factorization models* aim to decompose relational data into low-rank matrices for representation learning. Third, *Neural network models* encode relational data with non-linear neural activation and

more complex network structures [32]. At last, there are several models were proposed based on *GNN* [33, 22, 34], which are easy to learn the connectivity structure. As mentioned earlier, GNN-based KGC is regarded as an encoder-decoder format, and any one of the KGC models can perform the role of a decoder.

3.3.2 ConvKB

In DGC, we use the ConvKB [31] as a decoder, which performs the best among several models. ConvKB applies a convolution layer over the embedding triples (here each triple (*h, r, t*) is represented as a 3-column matrix where each column vector represents a triple element) [35]. This model keeps the transitional characteristic, and it makes great performance compare to other CNN KGC model [32]. Its score function is defined as

$$f(t_{ij}^k) = \left(\overset{\Omega}{\underset{m=1}{}}\mathrm{ReLU}([\vec{h_i}, \vec{g_k}, \vec{h_j}] * \omega^m)\right) \cdot \mathbf{W}. \tag{3.6}$$

ConvKB uses multiple filters to generate different feature maps; Ω denotes the number of filters. These feature maps generated by convolution are concatenated into a single vector to increase the learning ability of latent features. This single vector is then computed with a weight vector $\mathbf{W}$ via a dot product to give a score for the triple (*h, r, t*) (Fig. 3.1e). For training weight vector $\mathbf{W}$ of the model, we use Adam optimizer, minimizing the loss function $\mathcal{L}$ with L_2 regularization:

$$\mathcal{L} = \sum_{t_{ij}^k \in \{S \cup S'\}} \log\left(1 + \exp\left(l_{t_{ij}^k} \cdot f(t_{ij}^k)\right)\right) + \frac{\lambda}{2}\|\mathbf{W}\|_2^2$$
$$\text{in which } l = \begin{cases} 1, & \text{for } t_{ij}^k \in S \\ -1, & \text{for } t_{ij}^k \in S' \end{cases} \tag{3.7}$$

here S' is a collection of invalid triples generated by corrupting valid triples in S, just as written in Equation 3.5. The entire framework of DGC can be found in Figure 2.

Chapter 4

Experiments and Results

4.1 Dataset

We extract drug-drug and drug-protein interactions from DrugBank [36], and protein-protein interaction from HIPPIE [37]. We extract the golden standard set of DDIs including their quantified information from DrugBank. For drug-protein interactions, we only extract drugs which were included in the set of DDIs. For protein-protein interactions, we only consider proteins from HIPPIE that appear in the drug-protein interactions. The final network has 723 drug and 1578 protein nodes (total 2301 nodes) connected by 17674 drug-drug, 9363 drug-protein, and 4982 protein-protein edges (total 32752 edges). Table 4.1 shows DGC graph statistics, and we illustrate the details about each relation statistics in Section 1 of Appendix.

4.2 Method comparison

4.2.1 DDI Baselines

We compare the performance of our method with previous DDI studies in two aspects: one for a drug representation and the other for a state-of-the-art DDI prediction. For the first experiment, we choose baselines using the same feature set as used for DGC

but different embedding methods [2] and [38].

- [2] uses similarity-based embedding, which represents a drug by one-hot encoding, and applies similarity operations such as Jaccard or Tanimoto to represent drug pairs. This method is the most common way to represent drug pairs in DDI studies.

- [38] is similar to DGC in constructing a graph with drug and proteins. Still, they only consider whether drugs connect with proteins or not, not the type of relations, such as inhibit or induce. They generate a drug vector by node2vec [39] method and represent a drug pair by subtracting two vectors.

For DGC, we use a drug vector from an encoder part, and the way to represent a drug pair is the same as [38].

For the second experiment, we choose the most recent and high performance models, DDIMDL [18] and KGNN [17].

- DDIMDL uses four features of drugs: chemical substructures, targets, pathways, and enzymes. This study predicts drug interaction with related mechanisms and reports a performance by how each mechanism is well predicted. Of all kinds of mechanisms, a performance of two mechanisms was mainly considered; increased or decreased serum concentrations. For the final value, we averaged a performance of two mechanisms.

- KGNN constructs Knowledge Graph by collecting DrugBank data and using Bio2RDF tool. This method exploits topological information of each entity in KG and aggregates all neighborhoods to predict the potential DDIs. It implements multiple types of aggregations (concat, sum, neighbor) and classifies if drug pairs have interaction or not. [17] reports the performance by each aggregation method. Among them, we choose the highest value.

Other settings are applied as written in the original paper.

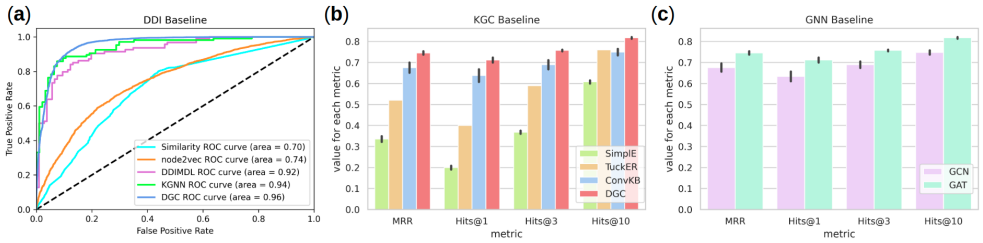

Figure 4.1: Performance evaluation of DGC on several challenging scenarios. (a) For DDI baseline, we got this result by conducting 5-fold cross-validation, in which the ratio between positive and negative samples is 1: 1. (b) For KGC baseline, a linear based model SimplE's value is 0.33, 0.2, 0.36, 0.6 (MRR, Hits@1, Hits@3, Hits@10 in order). A factorization based model TuckER's value is 0.52, 0.4, 0.59, 0.76 and a CNN based model ConvKB's is 0.67, 0.63, 0.69, 0.75. Our model's value is 0.74, 0.71, 0.75, 0.81. (c) For GNN baseline, GCN with relation's value is 0.67, 0.63, 0.69 and 0.75. DGC's value is the same as a KGC Baseline experiment. All results were summarized over five trials and expressed as mean $\pm$ SD.

In this experiment, every model gets drug pairs as input and decides they interact or not, i.e., conducts a binary classification task. We use AUC (Area Under the Curve) value is used as an evaluation metric. For the first experiment, all three methods use SVM (Support Vector Machine) as a classifier. The result of the first experiment presents our GNN based embedding method performs best under the same setting. As a result of the second experiment, DGC shows better performance than state-of-the-art models. Figure 4.1a shows the final result.

Table 4.1: DGC Statistics

#Entities	#Relations	#Triples	Train	Valid	Test
2301	9	32752	29052	1850	1850

4.2.2 KGC Baselines

We compare the performance of DGC with other Knowledge Graph Completion methods to see GNN-based KGC brings how much performance improvement. We use the best one of linear-based models (SimplE), factorization-based models (TuckER), and Convolutional Neural Network-based models (ConvKB) as the baselines. Table 4.2 shows score functions of baselines. Detailed explanations are follows:

- SimplE [40] is based on Canonical Polyadic (CP) decomposition in which head and tail entity embeddings for the same entity are independent. SimplE's scoring function alters CP to make head and tail entity embedding vectors dependent on each other by computing the two terms' average.

- TuckER [27] learns to embed by outputting a core tensor and embedding vectors of entities and relations. By having core tensor $\mathcal{W}$, TuckER does not encode all the learned knowledge into the embeddings; some is stored in the core tensor and shared between all entities and relations through multi-task learning.

We explain ConvKB in our model's decoder part.

In this experiment, every model got triple as input and returned score as output, which means the input triple's validity. We split DGC triples into train, valid, and test set as mentioned in Table 4.1. To evaluate each model's performance, we use the most common evaluation metric in KGC methods, a ranking procedure presented in [23]. First, for each triple in the test set, the head entity is removed and replaced by each entity of the entity set in turn to make a corrupted triple set. Then, scores of those corrupted triples are computed by the model's score function and then sorted by ascending order, which means a valid triple gets the lower score. Finally, the correct entity's rank is stored. This whole procedure is repeated while removing the tail instead of the head, and averaged values are reported. We report the value of metrics such as MRR (mean reciprocal rank), which calculates the mean of correct triple's rank in reciprocal and Hits@N, which the proportion of correct entities ranked in the top N

ranks for N = 1, 3, and 10. For both metrics, the higher value means better performance. We applied a filtered setting, which removes from the list of corrupted triplets all the triplets that appear either in the training, validation, or test set (except the test triplet of interest).

Figure 4.1b shows the result. The result clearly demonstrates that DGC significantly outperforms state-of-the-art results on four metrics. The performance difference between ConvKB and DGC also means the role of a Graph Neural Network. GNN improves performance by making it possible to predict the connectivity between the triples rather than simply using the KGC method.

Table 4.2: Score Functions of State-of-the-Art

Model	Score function
SimplE	$\frac{1}{2}(\vec{h_i} \circ \vec{g_k}\vec{h_j} + \vec{h_j} \circ \vec{g_k}'\vec{h_i})$
TuckER	$\mathcal{W} \times_1 \vec{h_i} \times_2 \vec{g_k} \times_3 \vec{h_j}$
ConvKB	$\left(\|_{m=1}^{\Omega} \mathrm{ReLU}([\vec{h_i}, \vec{g_k}, \vec{h_j}] * \omega^m)\right) \cdot \mathbf{W}$

Note: $\circ$ means Hadmard (element-wise) product, $\times_n$ denotes the tensor product along the n-th mode and $\|$ means concatenation.

4.3 Encoder comparison

We conduct this experiment to show our encoder GAT performs better than another GNN model, Graph Convolutional Network. We introduce the method using GCN [34] as a baseline that embeds entity and relation both, just like DGC. The convolution operation on graph can be summarized with two operations. First, in order to achieve a higher order representation of nodes, do a linear transformation paramterized by a weight matrix $\mathbf{W}$. The transformed feature vectors $\vec{h_i'}$ are given as $\vec{h_i'} = \mathbf{W}\vec{h_i}$. Then, to get the output features of node i, aggregate the features across the neighborhood of node. For relation embedding, [34] use several composition operators ϕ such as *sub-*

traction [23], *multiplication* [25] and *circular-correlation* [41]. Final feature vectors can be defined as:

$$\vec{h_i'} = \sigma\left(\sum_{(j,r)\in\mathcal{N}_i} \mathbf{W_r}\phi(\vec{g_k}, \vec{h_j})\right) \tag{4.1}$$

Because DGC utilizes TransE when initializing the relation vector, we choose *subtraction* operation for fair comparison.

We compared the performance by inserting the embedding vector created with the GCN into the input of the ConvKB. Experimental setting and performance metric are applied just the same as KGC baseline experiment. The result of comparison shows that DGC's performance increased by 8% to 12% (Figure 4.1c). While GAT implicitly captures the weight via an end-to-end neural network architecture so that more important nodes receive larger weights, GCN explicitly assigns a non-parametric weight to the neighbor during the aggregation process [42]. This difference makes GAT performs better as an encoder.

4.4 Case Study

Drug interaction studies concern about finding novel drug interaction pairs. Among the result of our model DGC, novel interaction pairs are validated by other databases. We use UpToDate [1] database, a software system that is a point-of-care medical resource. The validation target is the drug pair that returned the highest score but was not included in the train, valid, and test dataset, i.e., the pair that the model never observed. If UpToDate includes target pairs, we consider they are novel drug pairs. When searching for UpToDate, we paid attention to two points. First, if the type of interaction is PD, we do not include a drug pair in the novel pair set due to we limit our study's scope to PK interactions. Second, the quantification of interaction needs to be correctly predicted. When considering this, we additionally care about the order of drugs. For example, when (Fosaprepitant, increase, Eszopiclone) is a correct novel

[1] https://www.uptodate.com

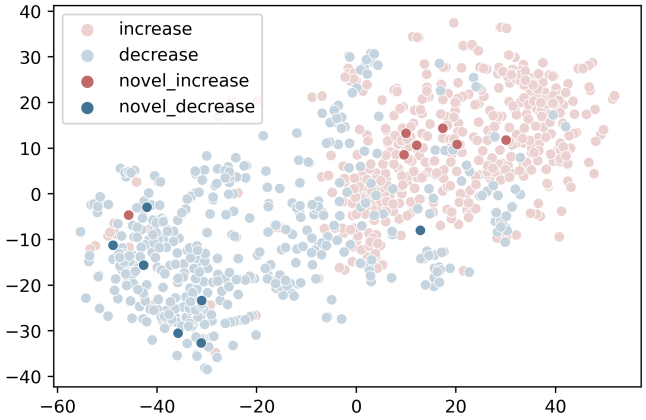

Figure 4.2: Sampled drug pairs are expressed in red for increase and blue for decrease. For novel pairs, we use darker colors.

pair, (Eszopiclone, increase, Fosaprepitant) would not be included in the novel pair set. The triple's head is a perpetrator drug, and the tail is a victim drug. When a prediction result is constructed oppositely, it is a wrong prediction. These constraints make novel pair validation more challenging compare to previous studies, which only consider a drug pair has interaction or not.

Table 4 shows the top 14 novel DDI pair predicted by the DGC. Seven novel pairs were verified for each increase/decrease case—these interactions were caused by inhibiting and substrating CYP enzymes, such as CYP3A4, CYP2D6, or CYP1A2. We embed drug pairs included in the training set and novel pairs together into a 2D space using t-SNE [43] and then visualize in Figure 4.2. Figure 4.2 reveals it DGC can classify quantification of interaction clearly. It also reveals the newly predicted novel pairs are included in each interaction cluster properly. This study shows DGC's ability to predict the quantification of interaction.

Table 4.3: Novel DDI pair Validation

Relation	Perpetrator	Victim	Summary
increase	Fosaprepitant	Eszopiclone	Fosaprepitant may increase the serum concentration of CYP3A4 Substrates (High risk with Inhibitors).
	Celecoxib	Thioridazine	CYP2D6 Inhibitors (Weak) may increase the serum concentration of Thioridazine.
	Simeprevir	Olaparib	Simeprevir may increase the serum concentration of CYP3A4 Substrates (High risk with Inhibitors).
	Fluvoxamine	Propranolol	CYP1A2 Inhibitors (Strong) may increase the serum concentration of Propranolol.
	Terbinafine	Flecainide	CYP2D6 Inhibitors (Moderate) may increase the serum concentration of Flecainide.
	Letermovir	Dronedarone	CYP3A4 Inhibitors (Moderate) may decrease the metabolism of CYP3A4 Substrates (High risk with Inhibitors).
	Nelfinavir	Amiodarone	Nelfinavir may increase the serum concentration of Amiodarone.
decrease	Magnesium hydroxide	Nilotinib	Antacids may decrease the serum concentration of Nilotinib.
	Apalutamide	Tipranavir	CYP3A4 Inducers (Strong) may increase the metabolism of CYP3A4 Substrates (High risk with Inducers).
	Fosphenytoin	Buspirone	CYP3A4 Inducers (Strong) may decrease the serum concentration of BusPIRone.
	Sevelamer	Calcitriol	Sevelamer may decrease the serum concentration of Calcitriol (Systemic).
	Orlistat	Ritonavir	Orlistat may decrease the serum concentration of Antiretroviral Agents.
	Rifampicin	Dronedarone	CYP3A4 Inducers (Strong) may decrease the serum concentration of Dronedarone.
	Phenytoin	Indinavir	CYP3A4 Inducers (Strong) may increase the metabolism of CYP3A4 Substrates (High risk with Inducers)

Chapter 5

Conclusion

This study proposes a drug interaction prediction model with a change of concentration by reflecting drug-protein relationships. DGC generates more informative drug embedding by Graph Attention Network and predicts which relation exists between two drugs by Knowledge Graph Completion. When conducting a comparative experiment of performance with existing DDI studies, we found that the embedding method proposed in this paper performs the best. In addition, a comparative experiment of link prediction methods shows our model, the Graph Neural Network-based Knowledge Graph Completion, showed the highest performance results compared to other Knowledge Graph Completion methods. Validation of novel drug pairs demonstrated that the model predicts both interaction happens or not and the interaction's quantification. There are some tasks we can do as future works. First of all, predicting the amount of concentration change would be more helpful to make therapy strategies. Second, as we present drug-protein relation in DGC graph, novel drug-protein interaction prediction to drug repositioning can be conducted without extra work. We can also apply various drug features to make specific predictions, such as genetic variation information or drug dosage.

Bibliography

[1] I. Segura-Bedmar, P. Mart?nez, and M. Herrero-Zazo, "Lessons learnt from the DDIExtraction-2013 Shared Task," *J Biomed Inform*, vol. 51, pp. 152–164, Oct 2014.

[2] F. Cheng and Z. Zhao, "Machine learning-based prediction of drug-drug interactions by integrating drug phenotypic, therapeutic, chemical, and genomic properties," *J Am Med Inform Assoc*, vol. 21, pp. e278–286, Oct 2014.

[3] C. B. Nemeroff, S. H. Preskorn, and C. L. DeVane, "Antidepressant drug-drug interactions: Clinical relevance and risk management," *CNS Spectrums*, vol. 12, no. S7, p. 1–16, 2007.

[4] F. Cheng, W. Li, G. Liu, and Y. Tang, "In silico ADMET prediction: recent advances, current challenges and future trends," *Curr Top Med Chem*, vol. 13, no. 11, pp. 1273–1289, 2013.

[5] B. Percha and R. B. Altman, "Informatics confronts drug-drug interactions," *Trends Pharmacol Sci*, vol. 34, pp. 178–184, Mar 2013.

[6] A. Gottlieb, G. Y. Stein, Y. Oron, E. Ruppin, and R. Sharan, "INDI: a computational framework for inferring drug interactions and their associated recommendations," *Mol Syst Biol*, vol. 8, p. 592, Jul 2012.

[7] R. Ferdousi, R. Safdari, and Y. Omidi, "Computational prediction of drug-drug interactions based on drugs functional similarities," *J Biomed Inform*, vol. 70, pp. 54–64, 06 2017.

[8] A. Kastrin, P. Ferk, and B. Lesko?ek, "Predicting potential drug-drug interactions on topological and semantic similarity features using statistical learning," *PLoS One*, vol. 13, no. 5, p. e0196865, 2018.

[9] S. Vilar, E. Uriarte, L. Santana, T. Lorberbaum, G. Hripcsak, C. Friedman, and N. P. Tatonetti, "Similarity-based modeling in large-scale prediction of drug-drug interactions," *Nat Protoc*, vol. 9, pp. 2147–2163, Sep 2014.

[10] W. Zhang, Y. Chen, F. Liu, F. Luo, G. Tian, and X. Li, "Predicting potential drug-drug interactions by integrating chemical, biological, phenotypic and network data," *BMC Bioinformatics*, vol. 18, p. 18, Jan 2017.

[11] P. Zhang, F. Wang, J. Hu, and R. Sorrentino, "Label propagation prediction of drug-drug interactions based on clinical side effects," *Scientific Reports*, vol. 5, p. 12339, Jul 2015.

[12] J.-Y. Shi, H. Huang, J.-X. Li, P. Lei, Y.-N. Zhang, K. Dong, and S.-M. Yiu, "Tmfuf: a triple matrix factorization-based unified framework for predicting comprehensive drug-drug interactions of new drugs," *BMC Bioinformatics*, vol. 19, p. 411, Nov 2018.

[13] H. Yu, K.-T. Mao, J.-Y. Shi, H. Huang, Z. Chen, K. Dong, and S.-M. Yiu, "Predicting and understanding comprehensive drug-drug interactions via semi-nonnegative matrix factorization," *BMC Systems Biology*, vol. 12, p. 14, Apr 2018.

[14] W. Zhang, Y. Chen, D. Li, and X. Yue, "Manifold regularized matrix factorization for drug-drug interaction prediction," *Journal of Biomedical Informatics*, vol. 88, pp. 90 – 97, 2018.

[15] M. Asada, M. Miwa, and Y. Sasaki, "Enhancing drug-drug interaction extraction from texts by molecular structure information," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Melbourne, Australia), pp. 680–685, Association for Computational Linguistics, July 2018.

[16] T. Ma, C. Xiao, J. Zhou, and F. Wang, "Drug similarity integration through attentive multi-view graph auto-encoders," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, p. 3477–3483, AAAI Press, 2018.

[17] X. Lin, Z. Quan, Z.-J. Wang, T. Ma, and X. Zeng, "Kgnn: Knowledge graph neural network for drug-drug interaction prediction," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (C. Bessiere, ed.), pp. 2739–2745, International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[18] Y. Deng, X. Xu, Y. Qiu, J. Xia, W. Zhang, and S. Liu, "A multimodal deep learning framework for predicting drug–drug interaction events," *Bioinformatics*, vol. 36, pp. 4316–4322, 05 2020.

[19] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, pp. i457–i466, 06 2018.

[20] S. Vilar, E. Uriarte, L. Santana, N. P. Tatonetti, and C. Friedman, "Detection of drug-drug interactions by modeling interaction profile fingerprints," *PLOS ONE*, vol. 8, pp. 1–11, 03 2013.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wal-

lach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 5998–6008, Curran Associates, Inc., 2017.

[22] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, "Learning attention-based embeddings for relation prediction in knowledge graphs," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (Florence, Italy), pp. 4710–4723, Association for Computational Linguistics, July 2019.

[23] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 2787–2795, Curran Associates, Inc., 2013.

[24] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web* (A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, eds.), (Cham), pp. 593–607, Springer International Publishing, 2018.

[25] B. Yang, S. W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, May 2015.

[26] T. Trouillon, J. Welbl, S. Riedel, E. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, p. 2071–2080, JMLR.org, 2016.

[27] I. Balazevic, C. Allen, and T. Hospedales, "TuckER: Tensor factorization for knowledge graph completion," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint*

Conference on Natural Language Processing (EMNLP-IJCNLP), (Hong Kong, China), pp. 5185–5194, Association for Computational Linguistics, Nov. 2019.

[28] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao, "Knowledge graph embedding via dynamic mapping matrix," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Beijing, China), pp. 687–696, Association for Computational Linguistics, July 2015.

[29] D. Q. Nguyen, K. Sirts, L. Qu, and M. Johnson, "STransE: a novel embedding model of entities and relationships in knowledge bases," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, (San Diego, California), pp. 460–466, Association for Computational Linguistics, June 2016.

[30] T. Dettmers, M. Pasquale, S. Pontus, and S. Riedel, "Convolutional 2d knowledge graph embeddings," in *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, pp. 1811–1818, February 2018.

[31] D. Q. Nguyen, T. D. Nguyen, D. Q. Nguyen, and D. Phung, "A novel embedding model for knowledge base completion based on convolutional neural network," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, (New Orleans, Louisiana), pp. 327–333, Association for Computational Linguistics, June 2018.

[32] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition and applications," 2020.

[33] C. Shang, Y. Tang, J. Huang, J. Bi, X. He, and B. Zhou, "End-to-end structure-aware convolutional networks for knowledge base completion," *CoRR*, vol. abs/1811.04441, 2018.

[34] S. Vashishth, S. Sanyal, V. Nitin, and P. Talukdar, "Composition-based multi-relational graph convolutional networks," in *International Conference on Learning Representations*, 2020.

[35] D. Q. Nguyen, "A survey of embedding models of entities and relationships for knowledge graph completion," 2020.

[36] D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda, N. Assempour, I. Iynkkaran, Y. Liu, A. Maciejewski, N. Gale, A. Wilson, L. Chin, R. Cummings, D. Le, A. Pon, C. Knox, and M. Wilson, "DrugBank 5.0: a major update to the DrugBank database for 2018," *Nucleic Acids Res*, vol. 46, pp. D1074–D1082, 01 2018.

[37] G. Alanis-Lobato, M. A. Andrade-Navarro, and M. H. Schaefer, "HIPPIE v2.0: enhancing meaningfulness and reliability of protein-protein interaction networks," *Nucleic Acids Res*, vol. 45, pp. D408–D414, 01 2017.

[38] S. Deepika and T. Geetha, "A meta-learning framework using representation learning to predict drug-drug interaction," *Journal of Biomedical Informatics*, vol. 84, pp. 136 – 147, 2018.

[39] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), p. 855–864, Association for Computing Machinery, 2016.

[40] S. M. Kazemi and D. Poole, "Simple embedding for link prediction in knowledge graphs," in *Advances in Neural Information Processing Systems 31* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 4284–4295, Curran Associates, Inc., 2018.

[41] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 1955–1961, AAAI Press, 2016.

[42] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020.

[43] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

Chapter A

Appendix

A.1 Node and Relation Statistics

We analyzed the node and relation information of DGC from various perspectives. This information would help to understand how drugs interact with other drugs or proteins.

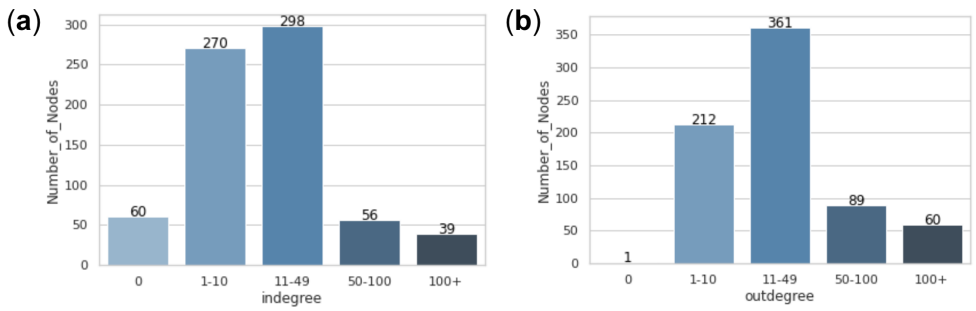

Figure A.1: The number of indegrees (a) means the number of perpetrator drugs and the number of outdegrees (b) means the number of victim drugs. Most drugs have 11 to 50 pair drugs which have interaction with.

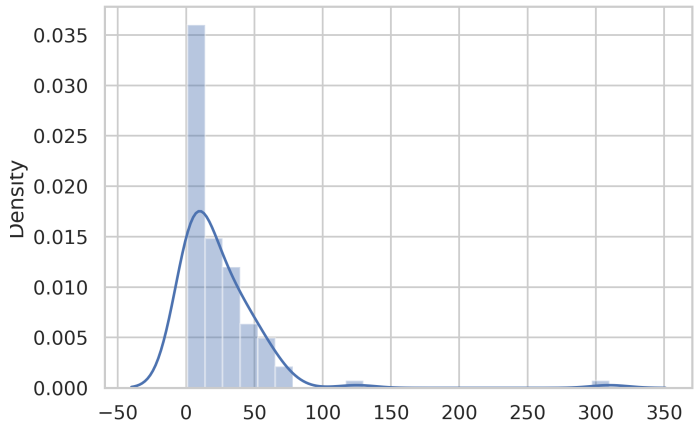

Figure A.2: Distribution about how much protein a drug is associated with. The drug with the most protein is *Fostamatinib*, which has a total of 310 proteins. On average, when we looked at 706 drugs, they interacted with 12 proteins. Figure S4 shows the distribution of the number of proteins each drug has.

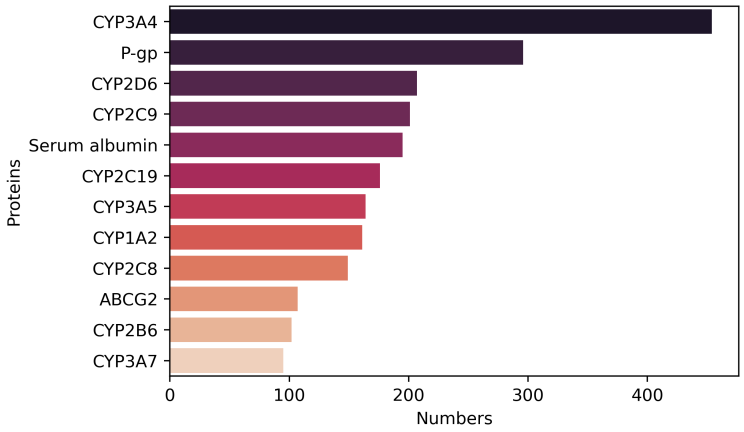

Figure A.3: The number of Top 12 Proteins. Most of them are CYP superfamilies (9 Enzymes), and the rests are a Carrier (serum albumin) and two Transporters (P-gp, ABCG2).

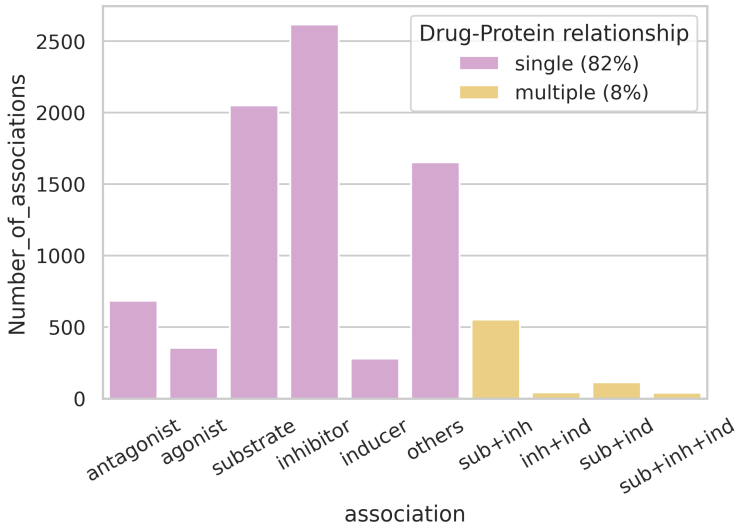

Figure A.4: We analyzed how many relations exist between one drug-protein pair. Of these, 82% relation was a unique interaction, and 8% was an interaction that occurred with other relation types

Table A.1: Relation Statistics Detail

Drug-Drug	increase	12287
	decrease	5387
Drug-Protein	antagonist	739
	agonist	373
	substrate	2775
	inhibitor	3220
	inducer	497
	others	1759
Protein-Protein	PPI	4982

Total number of DDI is 17674, Drug-Protein Interaction is 9363, and PPI is as written.

A.2 Ablation Study

We compare our model's performance with several experiment settings. First, we experimented with the performance according to the combination. Feature combinations including enzyme showed the highest performance.

Table A.2: Performance Differences by Feature Combination

	MRR	Hits@1	Hits@3	Hits@10
Target	0.77	0.74	0.78	0.83
Enzyme	**0.79**	**0.76**	**0.8**	**0.84**
Transporter	0.77	0.74	0.78	0.82
Carrier	0.75	0.72	0.76	0.81
Target + Enzyme	0.77	0.74	0.78	0.82
Target + Transporter	0.71	0.66	0.73	0.8
Target + Carrier	0.75	0.72	0.76	0.81
Enzyme + Transporter	0.76	0.72	0.77	0.83
Enzyme + Carrier	0.77	0.74	0.78	0.83
Transporter + Carrier	0.78	0.75	0.79	0.83
Target + Enzyme + Transporter	0.74	0.71	0.76	0.82
Target + Enzyme + Carrier	**0.79**	**0.76**	**0.8**	**0.84**
Target + Transporter + Carrier	0.78	0.74	0.79	0.84
Enzyme + Transporter + Carrier	0.73	0.7	0.75	0.79
Target + Enzyme + Transporter + Carrier	0.75	0.72	0.76	0.82

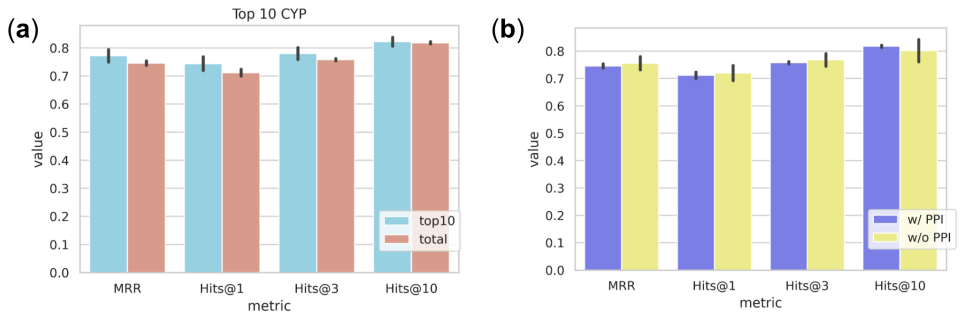

Figure A.5: (a) To confirm the effect of top 10 enzymes, we only include those proteins to the training set. The final graph includes 723 drugs and 10 proteins for nodes and 2568 drug-proteins and 1 ppis for relations. The values of top 10 CYP are 0.77, 0.74, 0.78 and 0.82 (MRR, Hits@1, Hits@3, Hits@10 in order) while including all proteins has 0.74, 0.71, 0.75, 0.81. (b) We also conduct experiments excluding Protein-Protein Interactions, others left still. The values of without PPI are 0.75, 0.72, 0.76, 0.82 while PPI including version's values are the same as including all proteins which are mentioned in (a).

A.3 The association of PPI and drug interactions

We organized a network without DDI information to see how much PPI information was related to the two drugs interacting. We experimented to see how many hops take to connect the drug pair which interact with each other. If the path connecting the drug pair is 2-hops, it means they share the same protein, and if it takes 3-hops, it means that the PPI connects the two drugs.

A.4 Performance on Other Databases

While DGC outperforms in Drug and Protein databases, Graph Neural Network-based Knowledge Graph Completion method shows great performance at other databases. We looked at representative Knowledge Graph, such as WN18RR, KB15K-237, NELL-995, and Kinship, which are WordNet or knowledge graph describes facts about movies, actors, awards, sports, and sports teams. The result with these knowledge graphs is presented in Table A.3. DGC performs best on kinship, and on other databases also performs better than other KGC methods. With this experiment, GNN based KGC is suitable for a multi-relational model to embedding and predicting links.

Table A.3: Experimental results

Database	MRR	Hits@1	Hits@3	Hits10
WN18RR	0.44	0.36	0.48	0.58
KB15K-237	0.51	0.46	0.54	0.62
NELL-995	0.53	0.44	0.56	0.69
Kinship	0.90	0.85	0.94	0.98

A.5 Parameter Sensitivity

In this section, we test the parameter sensitivity of DGC on our dataset and the results are presented in Figure A.6. We train DGC using a grid search of hyperparameters: embedding size $\in \{50, 100, 200, 400\}$ of encoder output, margin $\in \{1, 3, 5, 7\}$ for training encoder, weight decay $\in \{5e^{-04}, 1e^{-05}, 5e^{-05}, 5e^{-06}\}$ and epochs $\in \{1000, 2000, 3000, 3600\}$ used in learning encoder. The left-top figure shows the effect of embedding size when other parameters are fixed. The performance of DGC is best at embedding size is 100. Most evaluation metrics show a slightly lower performance at 200 and then a get higher performance again at 400. The right-top figure shows the

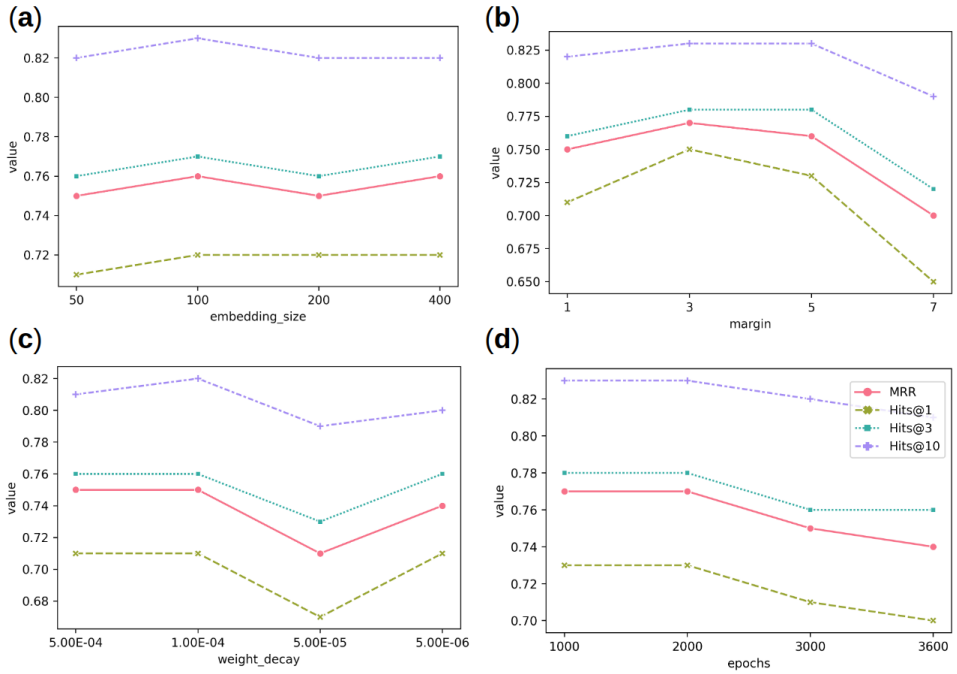

Figure A.6: Parameter analysis of DGC. Indicates the performance change of the four metrics according to the parameter change. As the epoch value changes, DGC showed the best performance in most indicators. Among the rest, the performance was higher in order of margin, embedding size, and weight decay.

effect of margin that corresponds to the value of γ in the loss function for training encoder. The performance of DGC goes down when $\gamma > 5$. The bottom figures show the result about parameters to use when learning an encoder model, GAT. As shown in Figure A.6c weight decay (λ) is best at $1e^{-04}$. Epochs best at 1000 or 2000, and as its value increases, it shows lower performance.

A.6 Result Analysis

We now analyze the result to interpret the predictive performance of our model. In the case of incorrect answers that did not return true triple to first place, see what charac-

teristics are shared between the drug predicted first and the drug actually correct. As Figure A.7 shows, if the correct answer triple is a drug pair in which CYP3A4 inhibits and substrates meet and interact, the incorrect response triple also has this feature. Drug *Imatinib* and *Asunaprevir* are two drugs that interact with CYP3A4. The result of head prediction for this triple was drug *Letermovir*, and the result of tail prediction was *Vilazodone*, CYP3A4 inhibitor, and substrate, respectively. As with an increase, decrease also calculated the highest scores of inducer and substrate drugs, resulting in the lower ranking of correct triplets. When P-gp inducer *Apalutamide* and P-gp substrate *Colchicine* are correct triple, the head *Tamoxifen* predicted as the highest score which is also P-gp inducer. Likewise, the tail *Levomilnacipran* predicted as the highest score is also P-gp substrate. Although the answer is incorrect, it can be seen that the result is due to these pharmacological characteristics.

In CYP3A4 or P-gp, recommending drugs that contain these proteins is possible because of the large number of drugs it has. By checking other protein examples, we showed that it was not a simple recommendation. In addition, for correct pairs that were not satisfied with the inducer-substrate relationship, it could be seen that the highest score was given to the pair that satisfies the inducer-substrate relationship. Drug *Gemfibrozil* and *Enzalutamide* have CYP2C9 in common but are in the inhibit - substrate relationship and have an interaction with decreasing concentration value. The DGC returned *Dabrafenib*, CYP2C9 inducer, to top-ranked in the head prediction. Figure A.7 illustrates these situations.

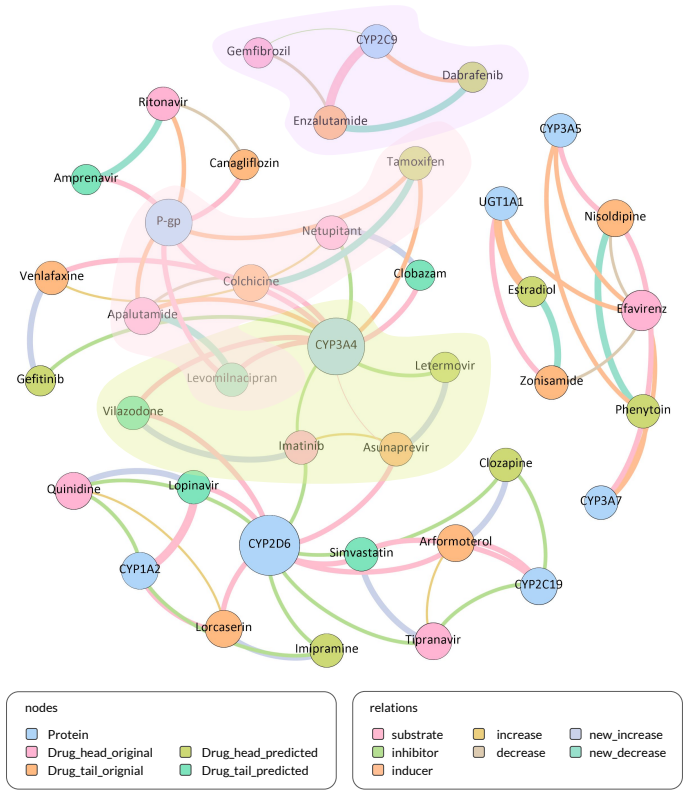

Figure A.7: A graph consisting of a total of 40 nodes and 75 relation. The size of the node means its degree. The legend of nodes and relation is described at the bottom of the graph. The drugs in the increase relationship were high-lighted in yellow, the drugs in the decrease relationship were red, and the drugs in which the index inhibit information was reversed with the connotation results were high-lighted in purple.

초 록

약물 상호작용 예측을 위해 여러 계산론적 방법이 제시되어 왔다. 타겟, 효소, 수송체, 운반체 등으로 분류되는 단백질은 상호작용의 원인이 되며 여러 계산론적 방법론에서 약물을 표현하기 위한 피쳐로 활용된다. 그러나, 기존의 약물 표현 방식은 충분한 정보를 포함하기에 한계가 있으며 두 약물간의 상호작용 발생 유무만을 판별할 뿐, 상호작용의 종류는 판별하지 못한다. 본 논문은 이러한 한계를 극복하기 위한 새로운 약물 표현 방식과 약물 상호작용 발생 정도까지 함께 예측하는 모델을 제안한다. 제안하는 모델은 각각이 인코더-디코더 역할을 하는 그래프 신경망과 지식 그래프 완성으로 이루어진 프레임워크이다. 먼저 그래프 신경망 중 하나인 그래프 어텐션 네트워크는 약물을 벡터로 나타내는 역할을 하는데, 이때 상호작용에 영향을 미치는 노드에 더 큰 어텐션 값을 부여하는 방식으로 동작한다. 다음으로 지식 그래프 완성 방법은 두 약물과 하나의 관계 벡터로 이루어진 트리플의 정당성을 계산함으로써 두 약물이 증가 혹은도 감소 방향으로 상호작용이 발생하는지 예측이 가능하게한다. 실험 결과를 통해 기존의 약물 상호작용 연구와 지식 그래프 완성 모델에 비해 제안하는 모델이 높은 정확도를 얻는 것을 보였다. 또한 모델이 예측한 새로운 약물쌍을 검증함으로써 상호작용 예측 시에 증가, 감소도 함께 예측이 가능한 모델이라는 것을 보이는 데 성공하였다.

주요어: 그래프 신경망, 지식 그래프 완성, 약물 상호작용
학번: 2019-22256

Acknowledgements

논문이 나오기까지 많은 분들의 도움이 있었습니다. 그분들께 감사인사를 전하고 싶습니다.

먼저 지도교수님이신 문봉기교수님께 감사인사 드립니다. 처음 연구실 들어온 때부터 연구실을 나서는 이 순간까지, 2년간의 석사 기간 동안 주신 많은 가르침 덕에 연구에 흥미를 가지게 되었고, 논문을 완성하는 힘을 길렀습니다. 늘 부족하지만, 앞으로의 자리에서도 최선을 다해 부끄럽지 않은 제자가 되도록 노력하겠습니다.

과제를 함께 진행한 서울대학교 융합과학기술원의 이형기 교수님과 서울대학교 약학대학의 오정미 교수님, 그리고 해당 연구실의 연구진들께도 감사인사를 전합니다. 그분들의 아낌없는 도메인 지식 공유 덕에 논문을 완성할 수 있었습니다.

우리 DBS 연구실 사람들에게도 감사인사를 전합니다. 랩장으로서 연구실 업무 전반에 신경 써준 찬호오빠, 필요한 순간마다 아낌없는 조언을 해준 보경오빠, 함께 과제 하면서 고생한 지현언니, 늘 바쁜 데도 어려움이 있을 때마다 잘 도와주던 교승이, 가장 비슷한 시기에 들어와서 비슷한 고민을 많이 했던 주훈오빠, 늘 늦게까지 연구실에 남아 자리를 지키던 철훈씨, 이런 저런 고민을 다정하게 잘 들어주던 온드라, 과제에 뒤늦게 합류해서 어려웠을 텐데 잘 따라준 재헌씨까지 모두에게 고맙습니다. 2년 동안 하고싶은 연구를 마음껏 할 수 있던 것도, 무사히 졸업할 수 있는 것도 모두 여러분 덕분입니다.

마지막으로 늘 힘이 되어준 가족들에게도 감사인사를 전합니다. 변함없는 내편이 있다는 것은 참 많은 힘을 가지는 것 같습니다. 물질적으로나 심적으로나 아낌없는 응원을 받은 덕분에 끝까지 지치지 않고 잘 마무리할 수 있었습니다. 코로나로 인해 여러모로 힘들고 어떻게 지나갔는지 모를 2020년이었지만, 저에게는 잊지 못할 한 해로 남을 것 같습니다. 도움주신 모든 분들께 다시 한 번 감사드리며 이 마음을 자양분 삼아 앞으로의 어려움도 잘 해쳐 나가도록 하겠습니다. 감사합니다.