Université de Tunis

Institut Supérieur de Gestion

Mastère de recherche

Spécialité

Sciences et Techniques de l'Informatique Décisionnelle (STID)

Option

Informatique et Gestion de la Connaissance (IGC)

Knowledge graph and attention-based drug-drug interaction prediction

Hadj Boubaker Rais Dalel

Soutenu le 04 Juillet 2023, devant le jury composé de:

Zied Elouedi	Professeur,ISG Tunis	Président
Meriem Jemel	Maitre assistante, ISG Tunis	Rapporteur
AFEF BAHRI	Maitre assistante, ESC Tunis	Directeur du méemoire

Laboratoire/Unité de recherche: Strategies for Modelling and ARtificial inTelligence Laboratory (SMART-LAB)

Abstarct

Identifying drug-drug interactions (DDIs) is one of the main concerns in drug discovery. Accurate prediction of potential DDIs is crucial for ensuring drug safety throughout the entire lifecycle of drugs. However, traditional experimental methods for discovering DDIs, are time-consuming, tedious, and expensive. Therefore, using computational methods with AI techniques has attracted many researchers in recent years. Yet, learning node latent embedding for DDI prediction is not well explored. In our work, we propose a novel model called Attention KGDNN-DDI, to resolve the DDI prediction. First, we learn drug representations from knowledge graph embeddings; next, we concatenate the learned drug embeddings and use an attention neural network to learn representations of drug-drug pairs; finally, we implement a deep neural network to accurately predict drug-drug interactions. Experimental results show that our method outperforms classic machine learning models.

Keywords: Drug-drug interaction, Deep learning, Attention mechanism, Knowledge graphs, Drug discovery.

Acknowledgments

Foremost, I would like to express my deepest gratitude to my advisor for her presence, advice, and continuous guidance to make this master's thesis possible.

I am also grateful to my family and my friends for the emotional support they have provided and for all the sacrifices that they have made on my behalf throughout my years of study.

Finally, I extend my gratitude to all the professors of the Higher Institute of Management of Tunis.

To my advisor, my family, my friends, and to all those who participated directly or indirectly in the development of this research work: because I owe it all to you. Many Thanks.

Contents

List of Figures

List of Tables

Abbreviations

ACC	Accuracy
AI	Artificial Intelligence
AUPR	Area Under the Precision-Recall Curve
AMF	Adjacency Matrix Factorization
AMFP	Adjacency Matrix Factor with Propagation
CNN	Convolutional Neural Network
DB	Database
DDI	Drug-drug interaction
DDI	Deep Neural Network
DL	Deep Learning
GNN	Graph neural networks
KG	Knowledge graph
KG	Knowledge graph embedding
LSTM	Long Short Term Memory
ML	Machine Learning
NLP	Natural Language Processing
NN	Neural Network
ROC-AUC	Receiver Operating Characteristic Area Under Curve
RNN	Recurrent neural network

Introduction

Motivation

Drug discovery plays an important role in today's society in treating and preventing sickness and possibly deadly viruses. During the initial stages of drug discovery development, the main challenge is to identify potential molecules to be used as drugs to treat a disease. The drug discovery and development of new drugs are notoriously costly and time-consuming. The typical process of a new drug can cost *US* $ 0.5-2.6 billions and takes 15 years [67]. The identification of interactions between drugs plays an important role in drug discovery [34]. Prediction of drug-drug interactions by computational methods can reduce research costs and shorten drug discovery time [21]. This makes developing efficient methods for predicting DDI prediction a vital and urgent task [33].

Drug-drug interactions (DDIs) prediction plays a crucial role in drug discovery, as they have the potential to significantly impact patient safety. DDIs occur when the effects of one drug are altered by the presence of another drug, leading to potential adverse reactions. Identifying and understanding these interactions is essential for healthcare professionals to make informed decisions regarding drug prescription and administration [33]. The prediction of drug-drug interactions (DDIs) plays a crucial role in minimizing the risk of adverse reactions and optimizing both the drug development process and post-marketing surveillance [56].

Traditional approaches of DDI prediction can suffer from drawbacks such as being time-consuming, high costs, and limited coverage. With the growing volume of electronic data and technology advancements, new opportunities have arisen to leverage these resources for more accurate and efficient DDI predictions. Many machine learning methods have been successfully applied in this task [33].

Some previous works have used the concept of drug similarities [4]. However, similarity-based approaches are unable to calculate various similarities for many drugs due to a lack of drug information.

Researchers have tended to devote less attention to the potential correlations between drugs and other entities, such as targets and genes. Additionally, recent studies also have adopted knowledge graphs for DDI prediction [42].

Those latter, have emerged as attractive methods for capturing and integrating complex relationships between entities. Using knowledge graph representation, each drug can have a large feature vector large enough to improve the classifier prediction power. One main advantage of choosing knowledge graphs is their ability to represent heterogeneous information contrary to single types of relations and the ability to continuously incorporate new data [5].

On the other hand, attention neural networks have shown great promise in learning from complex data and improving prediction accuracy. The idea behind attention mechanisms is to allow the network to focus its attention on specific parts of the input [76]. Yet, it has not been well explored in the context of DDIs prediction.

Based on this background, the aim of this thesis is to leverage the inherent knowledge graph structure and attention neural networks within the DDIs prediction task.

Objectives of the Thesis

The main objectives of the thesis can be stated as follows.

- Perform a literature review about the different ML-related approaches used for drug-drug interaction prediction.

- Develop a novel model based on attention mechanisms using the knowledge graph representation of drugs.

- Evaluate the performance of the developed model and compare the results with traditional machine learning models.

Thesis Outline

This thesis is composed of four chapters presented as follows:

- Chapter 1: This chapter provides an overview of knowledge graphs, deep learning models, and knowledge graph embeddings approaches.
- Chapter 2: The third chapter delves into drug discovery, its process, and the state of the art of drug-drug interaction approaches.
- Chapter 3: The fourth chapter presents the description of our proposed DDI prediction approach.
- Chapter 4: This chapter is dedicated to the experimental study and discusses the obtained results to evaluate and discuss our approach.

Finally, our master thesis concludes with a summary of the work presented throughout this report and suggests potential future possible improvements and perspectives of our approach.

Part I

Theoretical Aspects

Chapter 1

Background

Introduction

This chapter provides a brief overview of various background concepts and techniques that will be extensively utilized in this thesis. In what follows, an introduction to graphs and KGs is provided in sections 1.1 and 1.2 respectively. Next, Deep learning techniques are discussed in section 1.3, and lastly, Knowledge Graph Embeddings are represented in section 1.4.

1.1 Graphs

Graph Theory was first studied by the super famous mathematician *Leonhard Euler* in 1735 as a solution to the *Seven Bridges of Königsberg Problem* [27]. Graphs are everywhere. From social and communication networks to innovative bioinformatics research, to the World Wide Web, any system that consists of discrete states and connections between them can be modeled by a graph. Graphs serve as a mathematical representation of a network that is built upon to analyze and learn from real-world complex systems. A graph is a collection of objects, represented by nodes or vertices, along with a set of interactions between pairs of these objects which are connected by edges.

Graphs, denoted by $G = (V, E)$, are composed of vertices, $V = \{v_1, v_2, ..., v_n\}$, and edges, $E = \{e_{i,j}\}$. Each edge $e_{i,j}$ connects vertex v_i to vertex v_j. There are two common ways to represent graphs: using an adjacency matrix or a derived vector space representation. Graphs can be homogeneous graphs, heterogeneous graphs, with/without auxiliary [17].

- Homogeneous graphs are graphs in which all nodes and edges belong to the same type. In other words, each node and edge in the graph represents the same type of entity or relation, and there are no distinctions made between them [17].

- Heterogeneous graphs have different edge types to represent different relations among different entities and with directed or undirected edges. We could have them in community-based question-answering sites, multimedia networks, and knowledge graphs [17].

- Graphs with auxiliary information refer to graphs that include labels, attributes, node features, and information propagation. The labels indicate the type of each node. Attributes are discrete or continuous values that provide additional information about the graph beyond its structural information [17].

1.2 Knowledge graphs

The term ”knowledge graph” has been in use since 1972 [69], but its modern interpretation was popularized by Google through the announcement of the Google Knowledge Graph in 2012. Nowadays, there exist numerous knowledge graphs and applications of them. Some well-known open-source knowledge graph databases include DBpedia [8], FreeBase [12], YAGO [77], and Wikidata [86]. In addition, several enterprise or proprietary knowledge graphs are utilized by big corporations such as Microsoft, Google, eBay, and Facebook.

Knowledge graphs are commonly employed to store structured semantic data for various AI-related purposes [48]. The variety of applications for knowledge graphs is extensive, and they can be utilized in numerous specific use cases such as information retrieval [61], question answering [90], user recommendation systems [59], semantic search [43], chatbots [91], transport [89], and intelligent personal assistants [6].

Various papers have presented different definitions of knowledge graphs, but in this thesis, a definition proposed by Hogan et al [2] is used:

Definition 1.2.1. A knowledge graph is a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest whose edges represent potentially different relations between these entities.

Knowledge Graph (KG) is defined as $G = \{E, R, F\}$ where E, R and F are sets of entities, relations and facts, respectively. A fact is denoted as a triple (h, r, t) $\in F$ where h stands for 'Head', r stands for 'Relation' and t for 'Tail'. Another notation that is often used in literature is $< head, relation, tail >$.

Figure 1.1 illustrates an example of a KG extracted from DBpedia that consists of entities, relations, and literals. Here, we have `dbr:Stephan_Hawking`, `dbr:Oxford`, `dbr:University_of_Cambridge`, `dbr:Cambridge` are the entities. The entity `dbr:Stephan_Hawking` is of type `dbo:Scientist`, which is a subclass of `dbo:Person` representing the class hierarchy. The relations `dbo:birthPlace, dbo:almaMater, dbo:city` are object relations as they link two entities.

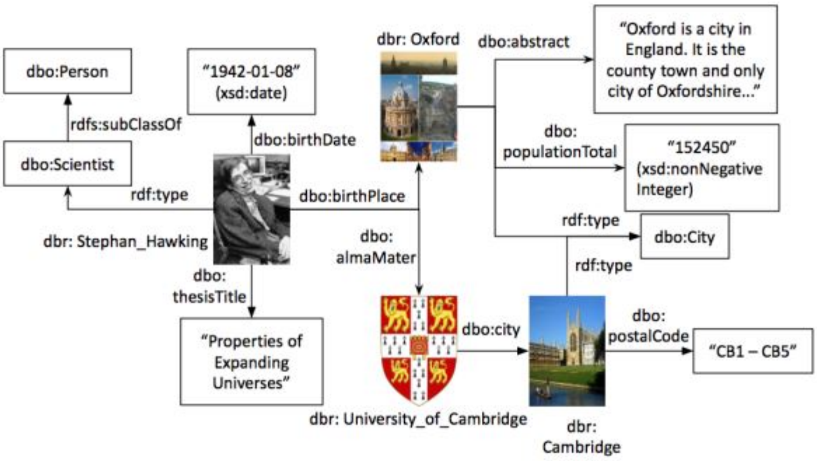


Figure 1.1: Knowlege graph extracted from DBpedia

Another example, in figure 1.2, illustrates three RDF graphs. We have four RDF molecules of types: `db:drugs, db:enzymes, dm:drugs, and dis:disease`, where prefixes `db` is for drugbank, `dm` is for dailymed, and `dis` is for diseasome datasets.

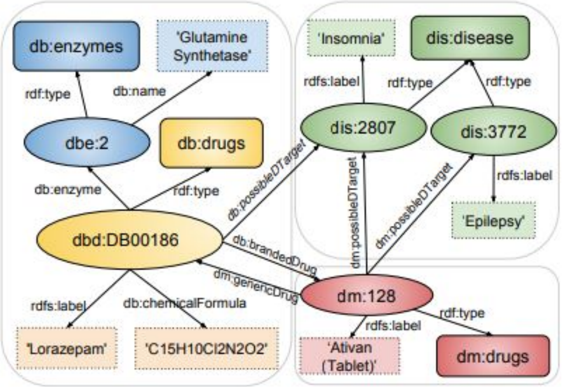


Figure 1.2: RDF graphs [26]

1.2.1 RDF

The Resource Description Framework (RDF) graph is a powerful data model widely utilized for information representation on the Web. It has been recommended by the World Wide Web Consortium (W3C) as a semantic web technology. The RDF framework enables the representation of information or facts regarding resources, which can be applicable to anything from articles and websites to people. The RDF model facilitates the deduction of facts based on these resources. These facts or statements consist of so-called triples.

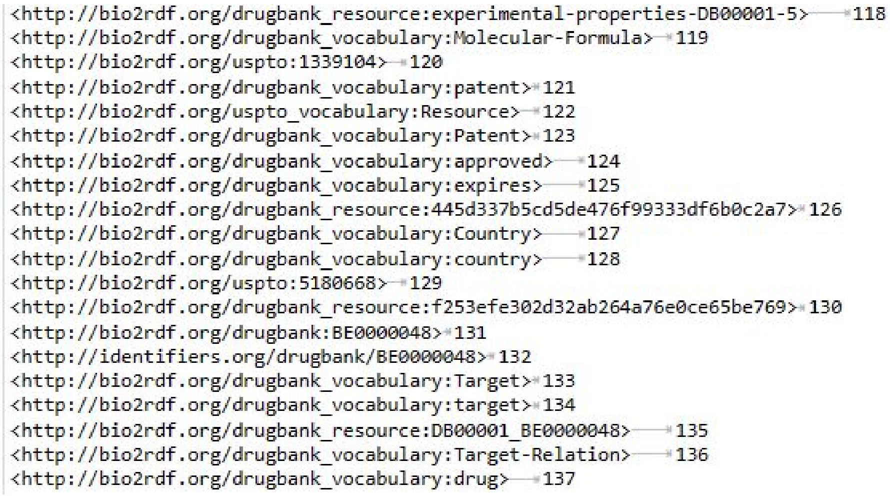
A graph in RDF is made up of triples that have a subject, predicate, and object. The predicate indicates the relationship between the subject and the object. The subject and predicate are always represented as URIs (Unique Resource Identifier), while the object can be either an URI or a literal (string, number, date) [28]. URIs are persistent, universal, and uniquely identifiable, making them suitable for annotating a diverse range of entities or concepts across various domains.

In table 1.1, there are several instances of RDF triples provided as examples. For instance, *(Lepirudin, hasTarget, prothrombin)* is a basic triple in a drug KG and indicates that there is a relationship hasTarget, linking *Lepirudin* to *prothrombin*. Additionally, figure 1.3 illustrates a variety of possible URI examples.

Linked Open Data (LOD) is a method that involves publishing and connecting structured data on the internet, making it easier to access and reuse information from various sources. This technique relies on Uniform Resource Identifiers

Table 1.1: RDF triples instances

subject	predicate	object
Lepirudin	hasTarget	prothrombin
Drug1	hasInteracted	Drug2
Disease2	presents	Symptom1
Drug2	hasSideEffect	SideEffect



```
<http://bio2rdf.org/drugbank_resource:experimental-properties-DB00001-5>    118
<http://bio2rdf.org/drugbank_vocabulary:Molecular-Formula>  119
<http://bio2rdf.org/uspto:1339104>  120
<http://bio2rdf.org/drugbank_vocabulary:patent> 121
<http://bio2rdf.org/uspto_vocabulary:Resource>  122
<http://bio2rdf.org/drugbank_vocabulary:Patent> 123
<http://bio2rdf.org/drugbank_vocabulary:approved>   124
<http://bio2rdf.org/drugbank_vocabulary:expires>    125
<http://bio2rdf.org/drugbank_resource:445d337b5cd5de476f99333df6b0c2a7> 126
<http://bio2rdf.org/drugbank_vocabulary:Country>    127
<http://bio2rdf.org/drugbank_vocabulary:country>    128
<http://bio2rdf.org/uspto:5180668>  129
<http://bio2rdf.org/drugbank_resource:f253efe302d32ab264a76e0ce65be769> 130
<http://bio2rdf.org/drugbank:BE0000048> 131
<http://identifiers.org/drugbank/BE0000048> 132
<http://bio2rdf.org/drugbank_vocabulary:Target> 133
<http://bio2rdf.org/drugbank_vocabulary:target> 134
<http://bio2rdf.org/drugbank_resource:DB00001_BE0000048>    135
<http://bio2rdf.org/drugbank_vocabulary:Target-Relation>    136
<http://bio2rdf.org/drugbank_vocabulary:drug>   137
```

Figure 1.3: Example of entities represented by URIs.

(URIs) to identify and interlink resources, resulting in an extensive network of interconnected data. The Resource Description Framework (RDF) is a crucial tool in LOD, providing a flexible and powerful way to represent and exchange data in a machine-readable format [39]. LOD's interconnected data can be utilized as a valuable source of background knowledge for machine learning, leading to more accurate and robust predictive models [39].

For example, Bio2RDF is a pre-existing linked open dataset that integrates a vast collection of databases in the life sciences domain accessible on different websites. It has developed a vast RDF graph that connects information from major biological databases associated with biological entities such as drugs, proteins, pathways, and diseases [54].

1.3 Deep Learning

With the availability of computational power and data, deep learning has become widely popular. Deep learning is a branch of machine learning focusing on artificial neural networks to learn and make predictions from large and complex datasets. Deep learning algorithms use multiple layers of artificial neurons to progressively extract features from the input data. These algorithms can be used for a wide range of tasks such as natural language processing [40], image recognition [57, 40] and autonomous driving [32].

One of the key advantages of deep learning is its ability to automatically learn features from raw data, without requiring features to be hand-engineered. This makes deep learning particularly useful for tasks where ML approaches may struggle due to the complexity of the data.

Through this section we are going to explore first neural networks as they are the backbone of DL. We will then get into LSTM networks, Convolutional Neural Networks, Attention mechanisms, GNNs and we will end with representing loss functions.

1.3.1 Neural networks

Artificial Neural Networks (ANNs) are made up of interconnected processing units referred to as Artificial Neurons, which are designed to imitate the functionality of biological neurons in the mammalian nervous system. *McCulloch* and Pitts pioneered the first computational model of neural networks in 1943 [49]. Then in 1958, *Rosenblatt* proposed the perceptron model which is a neural network for pattern recognition [64]. Neural networks offer a flexible approach to various problems by leveraging their ability to learn and extract patterns from data.

As shown in figure 1.4, the neuron has three fundamental components, a set of weights $(w_1, w_2..., w_n)$, a bias value (b), and an activation function. The weights and biases of the neuron are numerical parameters that are used to represent a linear combination of the inputs. Each input is multiplied by a weight followed by a summation of the results together with a bias and passed on to an activation function.

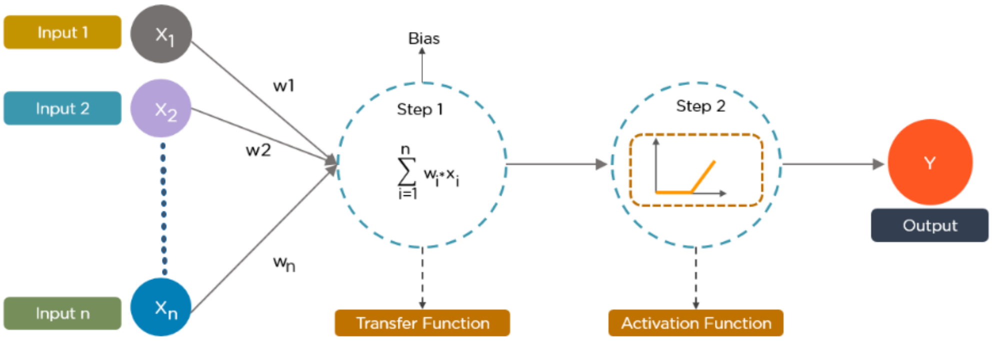


Figure 1.4: Neural Network Architecture [73]

This process is repeated throughout the network, with the output of one set of neurons serving as the input to the next set, until the final output is produced. This allows neural networks to learn complex patterns and relationships in data and can be used for tasks such as image recognition [57], and NLP [40].

There are three types of layers in a neural network: The input layer receives the input data, which is then processed and transformed before being transmitted to the output layer, and the hidden layers which act as intermediaries between the input and output layers.

Feed-Forward Neural Networks

A feed-forward neural network is made up of multiple layers of interconnected neurons that process input data [11]. The flow of information occurs in a unidirectional way, specifically from the input layer toward the output layer. [11].

Feed-back networks

In a Feed-back neural network, the connections are moving in both directions. Hence, the output derived from the network is given back to it, leading to improvements in its performance [94].

Activation functions

During the learning process, an activation function g(•) is employed to determine if a neuron should be activated based on the weighted sum calculation, which is further modified with a bias term. Below are various types of activation functions commonly used in neural networks.

1. Sigmoid:

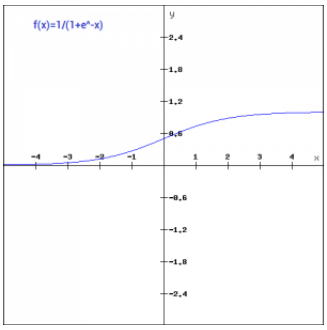


Figure 1.5: Sigmoid function

The sigmoid activation function maps input values to a range between 0 and 1. Its mathematical expression is given by the formula (1.1) :

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1.1}$$

2. Tanh:

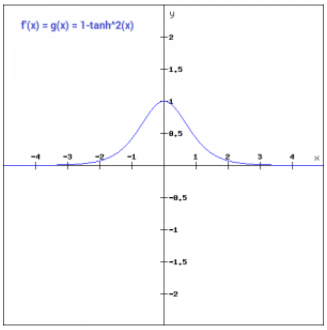


Figure 1.6: Tanh function

The tanh activation function is a nonlinear activation function. It differs from the sigmoid function with the particularity of being symmetric around the origin. The interval of values for the tanh function is from -1 to 1, which means that the output values can be negative or positive, depending on the input. The tanh function is mathematically defined as shown in (1.2):

$$tanh(x) = 2 \cdot \text{sigmoid}(2x) - 1 \tag{1.2}$$

3. ReLU :

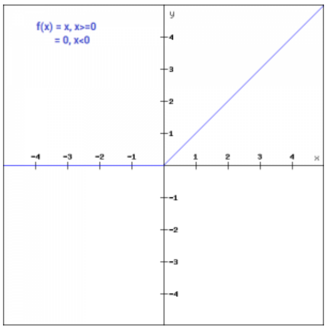


Figure 1.7: RELU function

The Rectified Linear Unit (ReLU) function is a type of non-linear activation function commonly used in deep learning.

It only activates the neurons whose input values are positive, while the neurons with negative input values are deactivated, which can improve the overall efficiency of the network. The mathematical expression for the ReLU function can be defined as 1.3:

$$f(x) = max(0, x) \tag{1.3}$$

1.3.2 Long Short Term Memory (LSTM)

Recurrent Neural Networks were first proposed by J.J. Hopfield [36]. In RNN, it is possible to use the output as an input for a previous layer or the current layer. For example, the output of a neuron at time t can be fed back as input to the same neuron or another neuron in the network at time t+1.

This type of feedback allows the network to maintain a memory of past inputs and computations and can be useful for tasks such as natural language processing [80].

For example, in voice recognition [74], we might wish to predict a phenome for every time step in an audio segment, based on past context.

However, RNNs suffer from several limitations, such as a limited range of contextual information and the back-propagation through time algorithm's inability to effectively handle the network's output, leading to what is known as the Vanishing Gradient Problem or the Exploding Gradient [35].

To overcome these limitations, Hochreiter and Schmidhuber [68] introduced in 1997, the Long Short-Term Memory (LSTM) network. The LSTM network was specifically designed to address these issues and is now widely used in a variety of applications. LSTMs deal with both Long Term Memory (LTM) and Short Term Memory (STM).

A basic LSTM network is composed of three primary components, that update and control the cell states, namely the forget gate, the input gate, and the output gate. By selectively opening or closing the gates, the LSTM network can regulate the flow of information and learn to store and recall information over extended periods [68].

As shown in figure 1.8, the network cell state is indicated by $c(t)$, while the output vectors transmitted through the network from time step t to t+1 are labeled as $h(t)$. The cell state vectors encode the relationships and dependencies over a long period, and it is the derivative of the cell state that can effectively prevent gradients from vanishing.

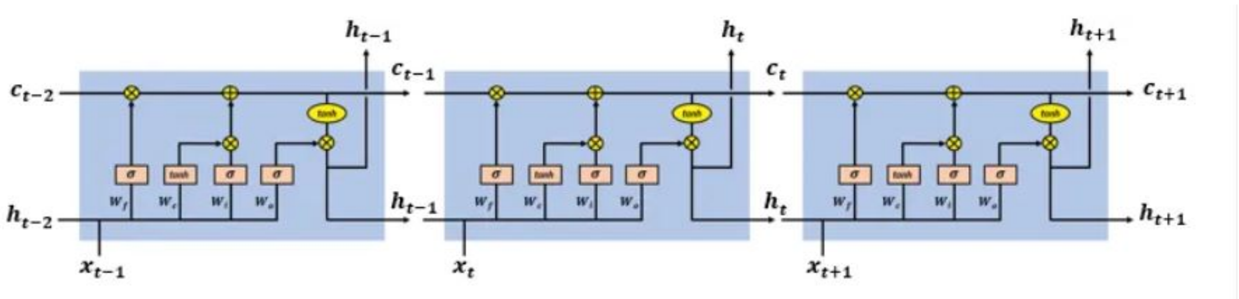


Figure 1.8: LSTM network cells at time steps t-1, t, t+1 [7]

1.3.3 Convolutional neural network (CNN)

CNNs are deep neural networks that are specifically designed to recognize patterns and features in 2-dimensional image data [37]. CNNs are commonly applied in image recognition [16], and they have been successfully used for a variety of other tasks. These include object detection [24], style transfer [30], and even natural language processing [74]. The ability of CNNs to learn complex patterns and features in data has made them a powerful tool in many different fields.

CNN Architecture

A CNN is made up of several stages, with each stage taking a set of feature representations as input and producing a new feature map. Each stage in the CNN has different layer types: a convolutional layer, a ReLU layer 1.3, and a pooling layer. In the end, a fully-connected layer will compute the class scores.

The convolutional layer connects perceptrons locally and preserves information about the neighboring perceptrons, processing them according to their corresponding weights [37]. This layer detects local combinations of features from the previous layer and maps them to a new feature representation. The pooling layer reduces the spatial dimensions of the feature maps, allowing the network to be more robust to variations in the position of features within an image. The most commonly used pooling techniques are MAX-pool and AVG-pool. Finally, the probability distribution over the different classes is computed through the fully connected.

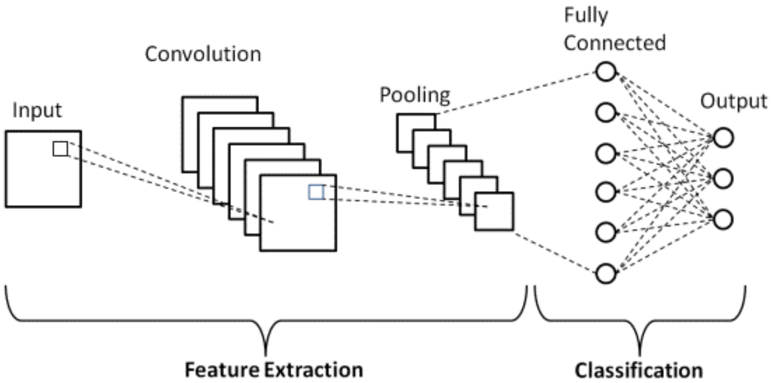


Figure 1.9: CNN architecture [58]

1.3.4 Attention mechanisms

For many years, researchers have dedicated their efforts for enhancing neural network architectures by incorporating attention mechanisms. This latter has been applied to various tasks, such as text classification [45], and recommendation [87]. Attention is a mechanism that lets neural networks concentrate on the most relevant parts of input or output for a given task. Neural networks, when processing sequences like text, speech, or images, can use attention to assign weights to certain elements, enabling a more relevant representation [53].

The year 2015 was a golden year for attention mechanisms. [10] proposed a simple but great idea for neural machine translation, where they suggested that not only can all the input words be taken into account in the context vector, but relative importance should also be given to each one of them.

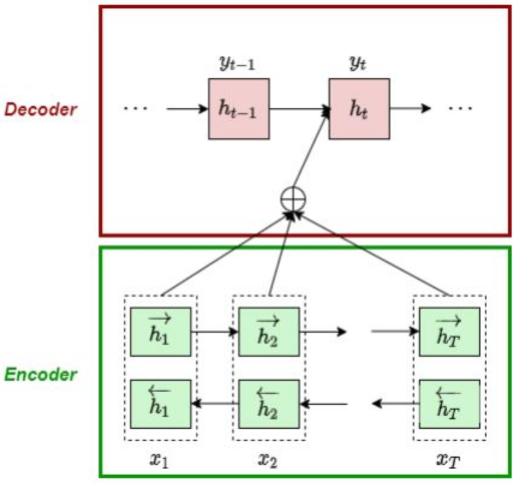


Figure 1.10: Illustration of a single step of decoding in an attention-based neural machine translation.The model tries to generate the *t-th* target word y_t given a source sentence $(x_1, x_2, ..., x_T)$ [76]

As shown in figure 1.10, the bidirectional RNN (BiRNN) takes word vectors as input, in the first part, the encoder. Then, the forward and backward states of BiRNN are computed. An annotation a_j for each word x_j is captured by concatenating these forward and backward hidden states. The Bidirectional LSTM produces a sequence of annotations $(a_1, ..., a_{T_x})$. The annotation a_j can emphasize the words around x_j because of the inherent nature of RNNs that represent recent inputs better.

In the decoder, a weight α_{ij} of each annotation a_j is achieved by using its associated energy e_{ij} that is computed using feed-forward neural network f as seen in (1.4). This neural network f is characterized as an alignment model that captures the relation between the inputs around position j and the output at position i. As it is shown in (1.5) α_{ij} is the output of a softmax function that determines the importance of annotation a_j with respect to the previous hidden state h_{i_1}. Finally, the weighted sum of these annotations is calculated as shown in (1.6) to determine the context vector c_i .

$$e_i j = f(h_{i-1}, a_j) \tag{1.4}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{1.5}$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} a_j \tag{1.6}$$

1.3.5 Graph Neural Networks

Graph Neural Networks (GNNs) is a class of Neural network used with data presented as a graph. A neural network is constructed based on the topology of the data graph. Nodes are connected to their neighbors as specified by the graph structure. The task of GNN is then to determine the node embedding for every node by taking into count the information on its neighboring nodes [2].

1.3.6 Loss functions

The loss function plays a crucial role in evaluating the performance of a deep neural network (DNN) by measuring how well the model predicted the correct outputs for the given input. It guides the optimization algorithm, to update the model's weights and improve its ability to fit the data. Here are some of the most used loss functions for a classification task.

Binary loss function

The binary loss function can be defined as follows:

$$\text{Binary Loss} = -\big(y \log(p) + (1 - y) \log(1 - p)\big) \tag{1.7}$$

where y represents the true binary label (0 or 1) of the sample, and p represents the predicted probability of the positive class.

Cross-Entropy Loss

The Cross-entropy loss can be defined as:

$$\text{CE} = -\sum_{i=1}^{N} \big(y_i \log(p_i) + (1 - y_i) \log(1 - p_i)\big) \tag{1.8}$$

The output of this loss function is a number between 0 and 1, where higher values mean a higher loss/error in the prediction.

1.4 Knowledge graph embedding

With the explosion of network volume, Knowledge Graph Embeddings (KGEs) offers a great solution to data sparsity problems by representing KGs in a low-dimensional space. KGE models are then capable to predict missing connections by learning embeddings of entities and relations and predicting a probability that a given triple is a fact.

Besides, those embeddings could be used as input to machine learning models. Those latter can benefit from the inclusion of domain knowledge. In fact, Knowledge Graphs (KGs) serve as a useful data structure for representing such domain knowledge. Since machine learning algorithms need inputs in numerical form, we can transform symbolic inputs into a numerical form through KGEs.

In this thesis the definition proposed by Wang et al [48] is used to define KGEs:

Definition 1.4.1. *Knowledge graph embedding is a technology for mapping the content of entities and relations in a KG to continuous low-dimensional vector space.*

The embedding dimensionality, denoted as d, is a fixed parameter generally chosen within the range of 50 to 1000. In graph embedding techniques, the embedding representation comprises two main components: entity embeddings and relation embeddings. For each entity (node) in the graph, an entity embedding, denoted as e, is generated. This entity embedding is a vector representation with d dimensions. Similarly, for each edge relation in the graph, a relation embedding, denoted as r is created. These vectors are used to abstract and retain latent structures in the graph.

A wide range of knowledge graph embedding techniques have been proposed. First, we examine translational models that are based on a geometric approach. Next, we will explore tensor decomposition models. Following that, we will delve into neural networks-based embeddings. Lastly, we will explore language models that exploit existing word embedding techniques.

1.4.1 Translation-based models

Translation-based models use the vectors of the subject node and the relation in order to predict the vector of the tail node. The relation r between a head entity h and a tail entity t is treated as a translation that maps the embedding of the head entity h to the embedding of the tail entity t, given a triplet (h, r, t), and $r \in R$ (the set of relations) and $h, t \in E$ (the set of entities).

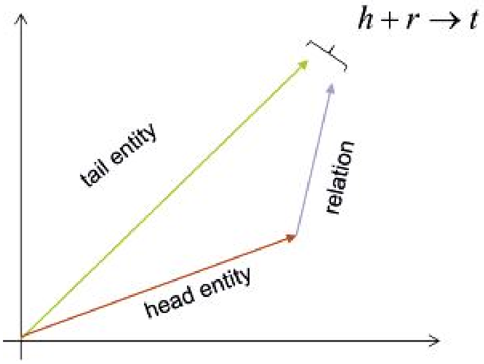


Figure 1.11: Translation-based embedding model. h, r, and t are embeddings of the head entity, relation, and tail entity, respectively.

TransE

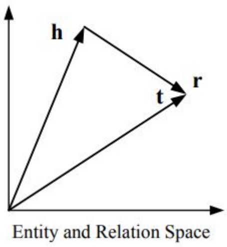


Figure 1.12: TransE

TransE [1] is an example of the translational family of KGE models, which follows the principle: $h + r \approx t$, so the vector of the head entity adds the relation vector is equal to the tail vector through a translation. This is achieved through some distance measure $d(h + r, t)$. Given a fact (h, r, t), the embedded entities h and t can be connected by r with low error. Figure 1.12 gives an illustration of this idea. The equation 1.9 refers to the scoring function which is defined as the (negative) distance between $h + r$ and t:

$$f_r(h, t) = -||h + r - t||^{1/2} \tag{1.9}$$

Over the training process, the TransE tries to minimize a margin-based loss function. The loss function is defined as follows :

$$L = \sum_{(h,r,t)\in S} \sum_{(h',r,t')\in S'} [\alpha + d(h + r, t) - d(h' + r, t')] \tag{1.10}$$

where S and S' indicate the set consisting of correct triplets and corrupted triplets, respectively, and Y indicates the margin hyper-parameter. During the training, TransE replaces the head or tail entity in each triplet with other candidate entities, resulting in the creation of a corrupted triplet set S'.

with $S' = \{(h_0, r, t) \mid h_0 \in E, (h_0, r, t) \notin S\} \cup \{(h, r, t_0) \mid t_0 \in E, (h, r, t_0) \notin S\}$

To better understand the TransE model, figure 1.13 illustrates the TransE approach in a real-world RDF graph where similar entities are close to each other in the identified cluster.

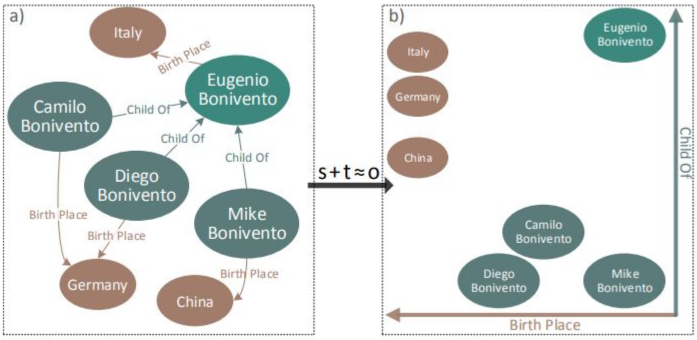


Figure 1.13: TransE Example

RotatE

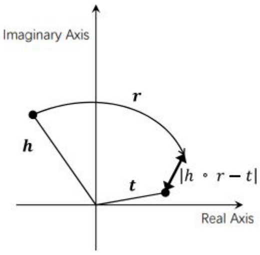


Figure 1.14: RotatE

RotatE [93] extends the notion of translation to rotation in the complex plane, making it possible to model the symmetry/antisymmetry, inversion, and composition patterns in knowledge graph relations. Therefore, given the triplet (h,r,t), the object vector representation is to be equal to t= s ⊙ t where ⊙ denotes the element-wise (Hadamard) product. The scoring function is defined as:

$$f_r(h, t) = ||h \cdot r - t||^2 \tag{1.11}$$

1.4.2 Tensor factorization-based models

Semantic matching models use similarity-based scoring functions to quantify the likelihood or plausibility of a triple in a knowledge graph. The main objective of these models is to capture and match the latent semantics of entities and relations, which are represented as vectors in a vector space.

By comparing the vector representations of entities and relations, the models can assess the similarity or compatibility between them. The scoring function used in the presented models is computed as a bilinear product:

$$f_r(h, t) = h \cdot r \cdot t \tag{1.12}$$

Some of the most known tensor decomposition models are Rescal [52], DistMult [9], and ComplEx [79].

Distmult

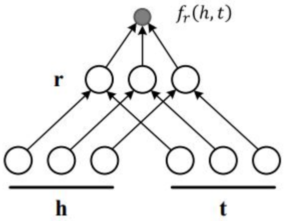


Figure 1.15: Distmult

DistMult employs a diagonal matrix to represent each relation in a way that the number of parameters grows linearly in terms of the embedding size. However, DistMult could not handle asymmetric relations, as (h, r, t) and (t, r, h) will be assigned to the same score.

For each relation r, it introduces a vector embedding $r \in \mathbb{R}^d$ and requires $M_r = \operatorname{diag}(r)$. The scoring function is hence defined as

$$f_r(h, t) = h^\top \operatorname{diag}(r) t = \sum_{i=0}^{d-1} [r]_i \cdot [h]_i \cdot [t]_i.$$

1.4.3 Neural network-based models

The use of neural network architectures enables embedding models to encode representations of knowledge graphs (KGs) into vector space with non-linear transformations.

ConvE

The ConvE as its name suggests uses a convolutional neural network (CNN) model. ConvE creates a 2-dimensional matrix by combining the subject and predicate vector representations, which are then fed into the convolutional layers. The output of these layers is a tensor of feature maps, which is transformed into a vector. This vector is then projected into a lower dimension and compared to the object vector representation using a dot product operation.

1.4.4 Language models

In natural language processing (NLP), a group of language modeling and feature learning approaches known as word embedding is used to map words or phrases from the lexicon to vectors of real numbers. Word2Vec, GloVe, and other neural network-based methods are trained on large corpora of text to generate word embeddings. That is words with similar context are assigned to similar vectors [2].

Skip Gram and CBow are two neural architectures for predicting word embeddings. CBow uses the central word to predict the surrounding words and Skip-gram Word2Vec predicts the neighboring words using the central one. GloVe utilizes a regression model over a matrix of co-occurrence probabilities of word pairs [2].

RDF2Vec is a method for creating vector representations of RDF graphs using Word2Vec. RDF2Vec first employs random walks on a given graph and saves the paths as sequences of RDF nodes. Then, those sequences are used as input into the word2vec model [62].

To better understand random walks, we consider $G(V,E)$ a graph consisting of a set of nodes V and a set of edges E. We start by generating all P_v paths from every vertex v in V and at a specified depth d. We employ the breadth-first algorithm. The algorithm initiates by traversing the direct outgoing edges of a root vertex vr. Subsequently, it randomly explores the connected edges through visited nodes until the specified number of iterations reaches d. The final set of sequences is formed by the union of all the P_{vr} walks, starting from all entities vr within the knowledge graph [62].

As shown in figure 1.16 once walks are generated, RDF2Vec uses one of the two following variants: a continuous bag of words (CBOW), to predict a word given its context words, or skip gram (SG), to predict the context words of a given word [62].

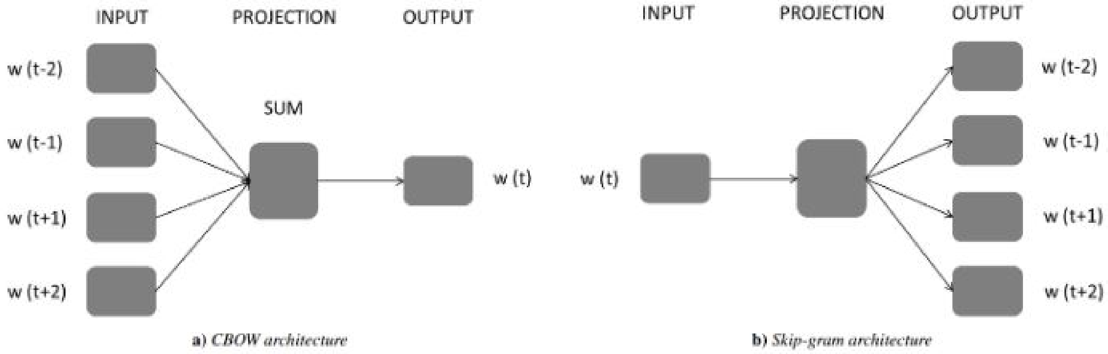


Figure 1.16: Architecture of the CBOW and Skip-gram model

Conclusion

This chapter has detailed the needed concepts related to knowledge graphs and deep learning, aiming for a better understanding of the rest of the thesis. In the next chapter, we will introduce the DDI prediction task and we will discuss its main related works.

Chapter 2

Related work

Introduction

This chapter gives a literature review associated with DDI prediction approaches. But first, it is important to introduce a comprehensive overview of drug discovery as well as the contribution of the use of knowledge graphs in this context. We will begin with an overview of drug discovery, in section 2.1. Next, we will describe the different KG applications for drug discovery, in section 2.2 In section 2.3, we will then focus on the drug-drug interaction prediction task and we will provide a state-of-the-art of the different used computational methods. Lastly, in section 2.4, we will end with the most commonly used databases for drug discovery.

2.1 Drug discovery

In simple terms, a drug can be defined as a molecule, or compound, that fits into a protein, called a target, in our body. This interaction between the drug and the target protein causes a subtle structure modification in the protein, inducing other chemical reactions [70]. These reactions are expected to have beneficial consequences for the organism's health.

Drug discovery and development play an important role in the diagnosis, treatment, and prevention of diseases. It is a multi-stage process from target identification to clinical trials [23].

New drug development is a time-consuming, tedious, and challenging process. The typical time from drug discovery to development is thought to be between 15 and 20 years [23]. Figure 2.1 represents the two main phases within the drug discovery and development process.

The first step in drug discovery is to decide which disease to study, the identification of the biological origin of a disease, and the potential targets. Then, the exploratory research starts. The Hit discovery involves screening millions of compounds to find activity when binding to a target. Then, the Hit to Lead process focuses on optimizing the hit to increase the strength of the binding. The next step involves further optimization to identify viable drug candidates. During this stage, researchers look for wanted properties, known as ADMET properties: absorption, distribution, metabolism, excretion, and toxicity. Once a promising compound is found, the pre-clinical phase will start. This involves the evaluation of the drug‘s safety and efficacy in animal species. Next, the clinical trials will be conducted on people, following a specific study protocol. Upon completion of the clinical trials, approval of the drug by the Food and Drug Administration is required to be given before the drug can be brought to the market [25].

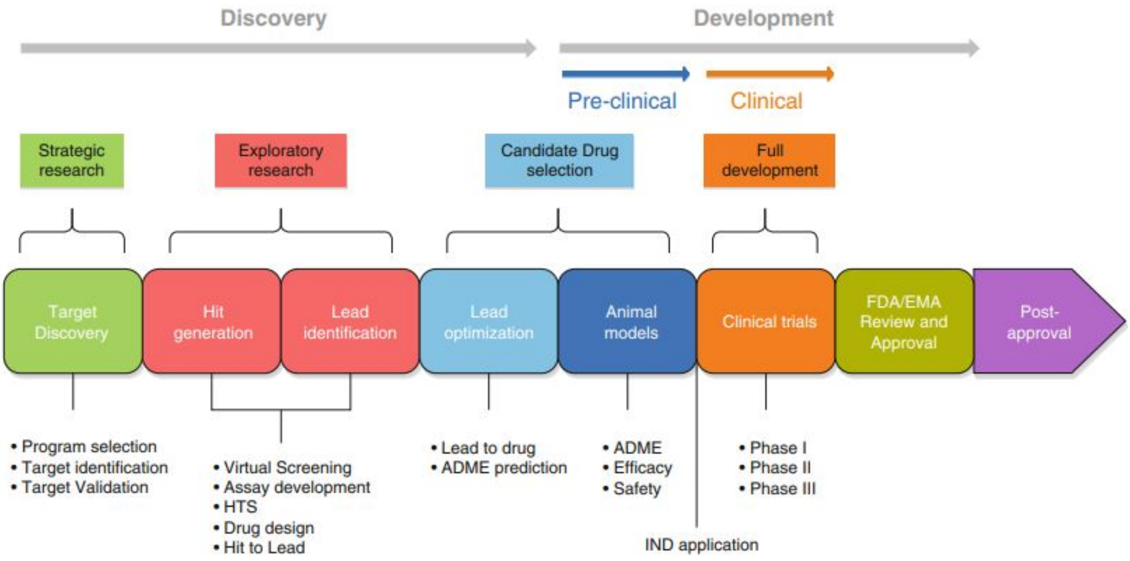


Figure 2.1: Representation of a new drug development process [25]

Term	Definition.
Biological System	A complex network that connects several biologically relevant entities.
Drug	A substance which, when administered to a living organism, can produce a beneficial biological effect.
Disease	A condition due to aberrant biological/physiological processes.
Target	A molecule or set of molecules that, when modulated by a drug, can influence or change the course of a disease.
Drug target	The specific receptor, enzyme, or cell on which the drug exerts its biological activity.
Hit	A compound that shows activity against the desired target when tested in a suitable assay, and which has had its structure confirmed.
Gene	Functional units of DNA, encoding RNA, and ultimately proteins. Variants of a gene's DNA sequence may be associated with disease(s).
Enzyme	A protein that acts as biological catalyst by accelerating chemical reactions.

Table 2.1: Main terms used within the scope of drug discovery

Main terms used within the scope of drug discovery:

Whilst a full review of the area is beyond the scope of this thesis, we tried to present an overview of the key concepts.

2.1.1 Subtasks of drug discovery process

The drug discovery process can be divided into smaller subtasks. Some of the most common subtasks can include [13]:

- Drug Repositioning: The goal is to identify potential treatments for novel diseases using existing drugs [13].
- Disease Target Identification: The goal is to find which molecular entities (genes and proteins) are involved in causing or maintaining a disease [13].

- Drug Target Interaction: Given a drug with unknown interactions, the goal is to find what proteins may interact within a cell [13].
- Drug Combinations: Identify the benefits, or toxicity that can arise from more than one drug being present and interacting with the biological system [13].
- Drug Toxicity Predictions: Identify the toxicities that may be produced by a drug. Those that arise from modulating the intended target of the drug and those that result from other properties or mechanisms of the drug

To reduce the cost and increase the success rate, researchers try to accelerate drug discovery by taking advantage of existing biomedical data. Researchers are particularly interested in leveraging predictions from drug-drug, drug-disease, and protein-protein interactions to expedite the process of drug discovery. Computational and machine learning approaches become more and more important in various tasks related to drug discovery [83]. Modern computational methods enable the identification of potentially beneficial or harmful interactions. As a result, drug trials can be narrowed down before actual clinical trials commence.

2.2 Application of KG in drug discovery

Lately, there has been an explosion of the data generated in the biomedical domain, and with rapid technological advancements, several graph methods have been emerging. When represented as a graph, data become useful for modeling biomedical entities and their relations [92]. Research has shown that the use of knowledge graphs is useful in several biomedical tasks. The use of KGs can facilitate the prediction of potential drug indications by leveraging drug-disease association graphs [31]. Also, disease-symptoms graphs could be used to support clinical decision-making [65].

Knowledge graphs (KGs) have become a valuable tool in the field of drug discovery, as they allow for the efficient representation of complex biological systems [13]. They have been utilized to represent biological data in various projects such as UNIPROT [20], Gene Ontology [19], and Bio2RDF [54]. These knowledge bases have served as the foundation for numerous predictive models that aim to identify associations between different biological concepts, such as drug adverse reactions and drug repurposing.

2.2.1 Graph representation

The task of learning biological associations is framed as link prediction in knowledge graphs. Predictive models leverage graph features and latent-space vector representations to infer a typed link between two nodes in the graph [51].

Graph features and latent-space vector representations are two different approaches for representing information in graphs:

Graph features models

The graph features models which fall under the network analysis methods, make predictions using various types of features such as random walks, network similarity, nodes connecting paths, and subgraph paths. These models are utilized in several biological predictive applications such as drug target prediction [55] and protein-protein interaction analysis [60]. While graph feature models are expressive in their predictions, they tend to focus on local graph features rather than global latent features as embedding models do.

Latent feature models

Entities and relations of KGs are represented through low-dimensional vector representations that maintain the graph's global structure [51].Encoding the local or global neighborhood of nodes in a graph into a discrete latent vector can then be used for downstream machine-learning tasks such as link prediction, and node classification. Furthermore, the embeddings generated from these KGs can also be used in unsupervised tasks such as community detection [51].

Embedding models have then been widely used in various applications, particularly in computational biology, for tasks such as predicting drug-target interactions (DTIs) [50] and forecasting drug polypharmacy side effects [97].

In a biomedical knowledge graph, drugs, genes, and diseases serve as entities, whilst the interactions occurring between them are encapsulated as relations, catching the associations between these entities. Therefore, facts are modeled as (subject, predicate, object) triples, e.g. (Aspirin, Drug–Target, COX-1), where a subject entity (drug) is connected to an object entity (target protein) through a predicate relation (Drug–Target).

As shown in figure 2.2, we have a knowledge graph that captures complex associations among various biological entities, including drugs, proteins, antibodies, and others.

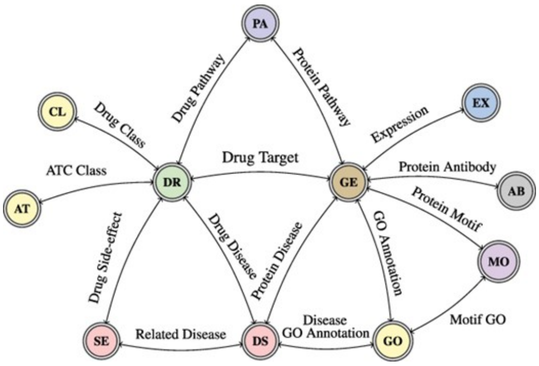


Figure 2.2: A knowledge graph modeling a complex biological system of different types of entities [51]

2.2.2 Link prediction

Link prediction in artificial intelligence is used to find missing links or discover future relationships that can occur in complex networks [3]. In drug discovery, various tasks are seen as predicting missing connections between entities. For example, drug repurposing can be thought of as the process of predicting connections that are currently missing between drug and disease entities. Similarly, target discovery involves the identification of missing links between genes and diseases, aiming to uncover potential associations.

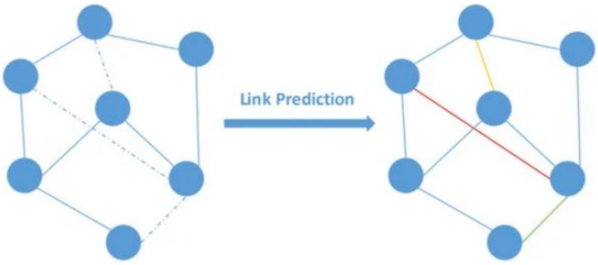


Figure 2.3: Link prediction

Here we present the major biomedical link prediction applications [3]:

- DDA prediction: Drug-disease association prediction task aims to predict which drug is associated with which disease.
- DTI prediction: Drug–target interaction prediction task is to predict which drug will affect which protein which is one of the applications in drug repurposing.
- DDI prediction: drug-drug interaction prediction aims to learn possible interaction between drug pairs.
- PPI prediction: Protein-protein interaction prediction learns possible interaction between protein pairs.

In this thesis, we focus on the drug-drug interaction prediction task.

2.3 Drug–drug interaction

When two or more drugs are combined together, there is a potential for drug-drug interactions (DDIs) to occur. The co-administration of medications can lead to drug-drug interactions, which can result in rising healthcare expenses [33].

Unfortunately, numerous DDIs are not discovered during the clinical trial phase, and they are only reported after the drugs are authorized for clinical use. These DDIs are known to cause adverse effects with serious consequences [33]. Therefore, it is imperative to detect or identify potential DDIs before administering or approving medications. Therefore, correctly predicting drug-drug interactions (DDI) will reduce adverse effects, but also will reduce drug development costs [33].

2.3.1 Input data for AI-DDIs studies

To keep up with the increasing number of pharmaceutical drugs introduced to the market in recent decades, several drug-related information databases have been updated and expanded to aid in the prediction of drug-drug interactions (DDIs). Previous studies on DDIs have typically utilized datasets from sources such as DDIExtraction 2011 [72], DDIExtraction 2013 [71], and the DrugBank database [88].

These publicly available sources offer a range of drug characteristics and DDI events that can be leveraged for AI-based DDI discovery. Hence, the available source for AI-DDI studies could be divided to [84]:

- DDIs information retrieved from text-based sources, which involve extracting information on drug-drug interactions (DDIs) from biomedical text, particularly scientific literature.
- Molecule-based input data and feature pre-processing for DDIs prediction: which use the information on chemical, molecular, and pharmacological properties to gain insights into drug interactions. Also, the knowledge graphs-based features integrated from multiple sources such as DrugBank [88], PharmGKB [82], and KEGG drugs [41] can be used to overcome the limited information issue in single-source methods.

  The overall workflow of traditional ML and DL for DDIs prediction is illustrated in figure 2.4.

2.3.2 DDI prediction methods: State of the art

Researchers have introduced a lot of computational methods to accelerate the prediction process of drug-drug interaction.

The existing methods for predicting Drug-Drug Interactions (DDIs) using computational approaches can be broadly classified into five different categories Similarity-Based Approaches, Factorization-Based Approaches, Literature-based Approaches, Deep Learning-based Approaches, and Ensemble Approaches.

Hence, we tried to represent them in this table 2.2

- **Similarity-Based Approaches**

  The main idea of similarity-based approaches is that similar drugs could interact with the same drug. That is if drug A and drug B produce a specific effect when taken together, then other drugs similar to drug A or drug B could also produce the same effect when taken with the other drug.

  For example, Tiresias [4] is a framework that utilizes the semantic integration of data to construct a knowledge graph. The knowledge graph is then used to

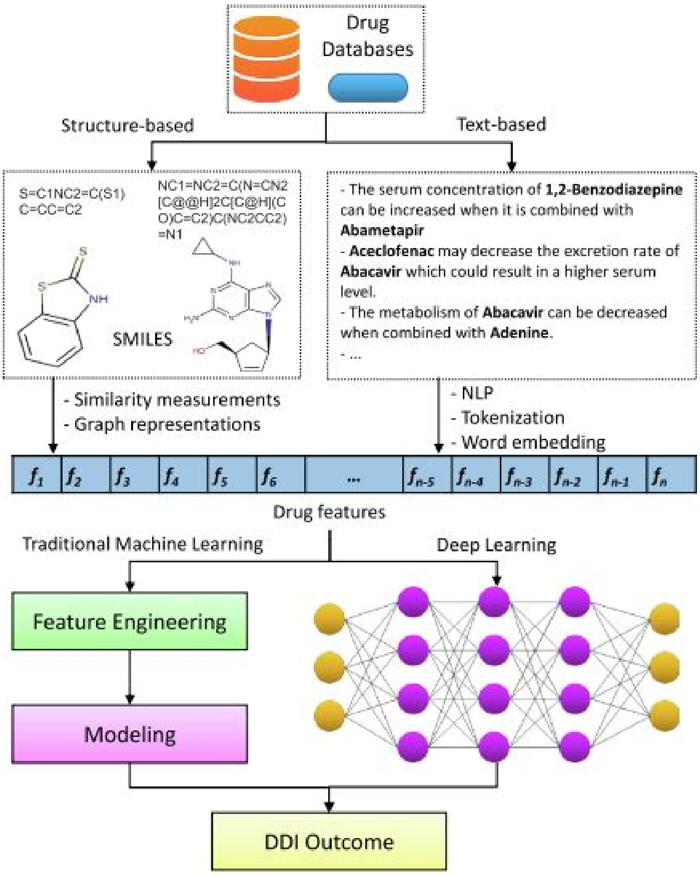

Drug Databases
Structure-based
Text-based
S=C1NC2=C(S1)C=CC=C2
NC1=NC2=C(N=CN2[C@@H]2C[C@H](CO)C=C2)C(NC2CC2)=N1
S
S
NH
NH
N
N
N
N
H_2N
HO
SMILES
- The serum concentration of **1,2-Benzodiazepine** can be increased when it is combined with **Abametapir**
- **Aceclofenac** may decrease the excretion rate of **Abacavir** which could result in a higher serum level.
- The metabolism of **Abacavir** can be decreased when combined with **Adenine**.
- ...
- Similarity measurements
- Graph representations
- NLP
- Tokenization
- Word embedding
f_1 f_2 f_3 f_4 f_5 f_6 ... f_{n-5} f_{n-4} f_{n-3} f_{n-2} f_{n-1} f_n
Drug features
Traditional Machine Learning
Deep Learning
Feature Engineering
Modeling
DDI Outcome

Figure 2.4: Overall workflow of traditional ML and DL for DDIs prediction [84]

calculate similarities among drugs. The produced similarity metrics are used to construct features for a logistic regression model.

In another work, the similarity of drug pairs was determined using the Russell-Rao method, which involves measuring the similarity of 12 binary vectors. A higher degree of similarity indicates a higher likelihood of drug interactions.[29]

- **Factorization-Based Approaches**

  Matrix factorization-based methods decompose the known drug-drug interaction (DDI) matrix into multiple potential matrices, which are then reconstructed to create a new interaction matrix. For example, in [75] a method for predicting drug-drug interactions (DDI) was proposed using two techniques: adjacency matrix factorization (AMF) and adjacency matrix factorization with propagation (AMFP). The input data for the prediction model only consists of known DDI. AMF decomposes the DDI matrix into an ad-

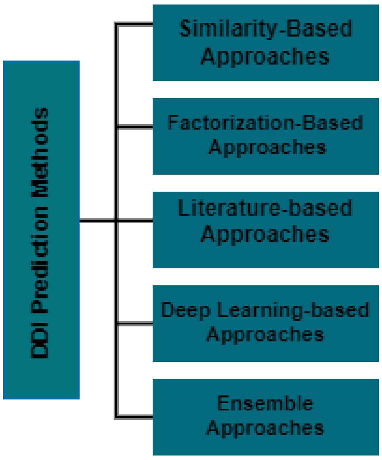

DDI Prediction Methods
Similarity-Based Approaches
Factorization-Based Approaches
Literature-based Approaches
Deep Learning-based Approaches
Ensemble Approaches

Figure 2.5: DDI prediction methods

jacent matrix, whereas AMFP extends the AMF version. AMFP propagates the factors of each drug to its interacting drugs.

- **Literature-based Approaches**

  Those methods consist of two main steps: First, relevant data is extracted from unstructured sources like scientific literature, and medical reports using text-mining methods and NLP techniques. Then, machine learning algorithms are used to predict unknown DDIs. For example in [78] an embedding mechanism was applied to obtain the semantic and syntactic information of the original literature. Then, a novel architecture was proposed to produce embedding-based convolutional features. Finally, these features are used as input to a softmax classifier to extract DDIs.

  [14] proposed a method for identifying likely drug interactions by constructing a knowledge graph from literature specific to diseases. The graph was created using natural language processing, indexing of biomedical publications, and open-source resources. The relationships between drugs in the graph were extracted and transformed into feature vectors that represent drug pairs, to facilitate the discovery of potential interactions.

- **Deep Learning-based Approaches**

  DeepDDI [66] is considered an early DDI (drug-drug interaction) prediction model that uses the chemical substructure similarity of drugs as input and predicts the interaction type using a Deep Neural Network (DNN). This model is notable for its ability to predict both single-type and multi-type

Table 2.2: DDI prediction related papers

Authors	Year	Input Data	Task	Feature Representation	Prediction	Metrics
Karim et al. [42]	2019	Integrated knowledge graph from Drug-Bank,PharmGKB, and KEGG	DDI	RDF2Vec, SimpleIE, TransE, KGloVe, CrossE , ComplEx	Convolutional-LSTM network	F1-score= 0.92
Ryu et al. [66]	2018	Drug–drug or Drug–food pairs and their structural information	DDI and DFI	Structural similarity profile (SSP)	DNN	Accuracy = 0.924
Rohani N, Eslahchi C. [63]	2019	drug similarities based on drug substructure, target, side effects, side effects, pathway	DDI	integrated similarity matrix	ANN	AUC from 0.954 to 0.994
Deepika et al [22]	2018	DDI network	DDI	node2vec	meta classifier	F1-score = 0.9506, ROC-AUC = 0.9912
Xuan Lin et al. [44]	2020	KG	DDI	-	Knowledge Graph Neural Network	F1-score = 0.9506, ROC-AUC = 0.9912
Shanwen Zhang et al. [96]	2022	KG	DDI	HolE model	Bi-LSTM with attentionm	Precision = 0.7425
Thanh Hoa Vo. [85]	2023	Drug Chemical information	DDI	Molecular fingerprints	Ensemble of Random Forest, XGBoost, and deep neural network	ACC= 0.9380
Zhang et al. [95]	2022	Multiple drug features	DDI	Jaccard similarity	CNN	AUPR = 0.9251
Shtar, G et al. [75]	2019	DDI graph	DDI	Adjacency matrix	XGBoost ensemble-based classifier	AUROC =0.814
Joshi, P et al. [38]	2023	KG	DDI	Node2vec	DNN	AUC-ROC =0.917

DDIs and has been used as a benchmark for comparison in various DDI prediction studies.

CNN-DDI method uses multiple drug features (pathways, enzyme, target, category), calculates Jaccard similarity of drugs' features, and trains a CNN model to predict DDIs [95].

[46] proposed a DANN-DDI model which takes drug features as drug feature networks, then the Structural Deep Network Embedding is utilized to learn the different drug embeddings from these networks. Once concatenated an attention neural network and a deep neural network are used to predict DDIs.

Knowledge graph-based methods

Further works have combined KGs and DL to predict DDIs, as shown in figure 2.6, [42] has combined CNN and LSTM layers in order to catch important features from knowledge graph embedding. The authors have applied RDF2Vec, SimpleIE, TransE, KGloVe, and CrossE to get representations of the nodes in the KG. Then the representations are provided to Conv-LSTM for prediction where the CNN is employed to capture local relationships while the LSTM is used to capture global ones.

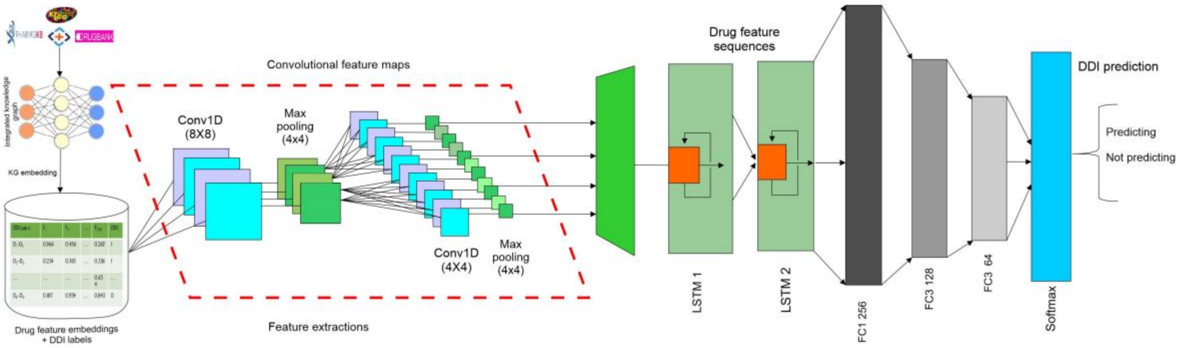


Figure 2.6: Conv-LSTM for DDI prediction [42]

In [44] a new end-to-end framework called KGNN was proposed and designed to capture drug-related information and their neighborhood relationships from a knowledge graph. In this framework, the authors utilized a Graph Neural Network (GNN) to learn representations of drugs from the knowledge graph by computing entity representations and aggregating neighborhood representations. As you can see in the figure 2.7, we have a schematic representation of the Conv-LSTM network, starting from taking input into an n-dimensional embedding space and passing to both CNN and LSTM layers before getting the vector representation of the most important features to fed through dense, dropout, Gaussian noise, and Softmax layers for predicting possible drug-drug interactions.

Another work presents a novel method for predicting Adverse Drug Reactions (ADRs) using a combination of Knowledge Graph (KG) embedding and a custom Deep Neural Network (DNN) called KGDNN (Knowledge Graph DNN). The authors begin by constructing a Knowledge Graph with six types of entities: drugs, ADRs, target proteins, indications, pathways, and genes. Each entity is embedded into a feature space using the Node2Vec algorithm. These embeddings capture the relationships and connections between different entities in the KG. The KGDNN model is then trained using these embeddings to classify ADRs [38].

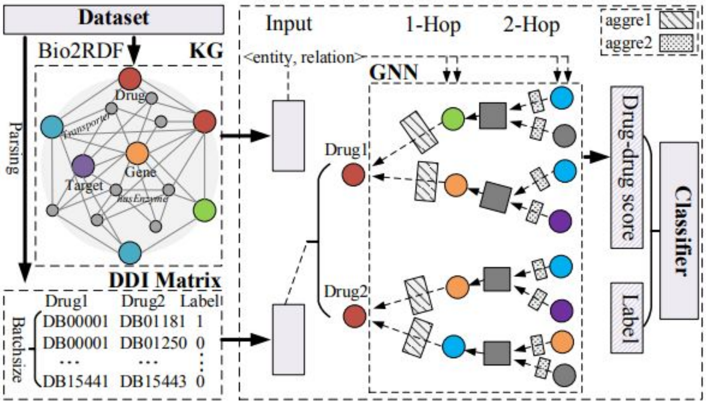


Figure 2.7: KGNN overview [44]

Besides, more recent work has emerged [96], the proposed approach for predicting drug-drug interactions (DDI) involves combining a knowledge graph (KG) with a Bi-directional Long-Short-Term Memory network (Bi-LSTM) that incorporates attention. The KG is utilized to integrate multiple sources of DDI information, which are transformed into vectors using the HolE knowledge graph representation model. The Bi-LSTM is then employed to extract implicit features of DDI, and a Softmax classifier is applied to identify DDI.

- **Ensemble Approaches**

In[22], the authors predicted drug-drug interactions (DDIs) using a combination of Node2vec, as a network representation learning method and a PU learning-based classifier called the bagging SVM classifier.

A more recent work, [85] proposed a novel model for predicting drug-drug interactions, by stacking three optimal models: Random Forest, XGBoost, and deep neural network.

2.4 Drug related databases

Table 2.3: Commenly used databases for biomedical knowledge graph construction

Data	Speciality	Description	
Drugbank	Drug	Drug interactions, pharmacology, chemical structures, targets and metabolism	*https : //go.drugbank.com/*
TWOSIDES	Side effects	Drug effects and drug-drug interaction side effects	*https : //tatonettilab.org/offsides/*
PharmaGKB	Gene	Genotype, molecular, and clinical knowledge	*https : //www.pharmgkb.org/*
KEGG	Gene	Genomes, biological pathways, diseases, drugs, and chemical substances	*https : //www.genome.jp/kegg/*

In recent years, many researchers and pharmaceutical companies have been building biomedical databases to address various drug discovery challenges [13], such as DrugBank, TWOSIDES, KEGG, and PharmKB databases.

DrugBank DrugBank is an online and free-to-use database. It was first released in 2007 [88]. It is a rich source of drug-target information and drug properties. The latest release of DrugBank (version 5.1.10, released 2023-01-04) contains 15,758 drug entries.

TWOSIDES TWOSIDES is the only comprehensive database of drug-drug-effect relationships. It contains over 3300 drugs and has 63000 combinations connected to millions of probable adverse effects [81].

PharmGKB The Pharmacogenomics Knowledge Base is an interactive tool for researchers investigating how genetic variation affects drug response. PharmGKB contains a variety of information on drugs, including their chemistry, and how they interact with different biological pathways. It contains more than 20000 genes, more than 3000 diseases, more than 2500, and 53 pathways [82].

KEGG The KEGG (Kyoto Encyclopedia of Genes and Genomes) was first developed in 1995. At that point, it only consists of four databases, PATHWAY, GENES, COMPOUND, and ENZYME and KEGG pathway. Then, the KEGG DRUG and DISEASE databases were introduced in 2005 and 2008, respectively.

The KEGG DRUG contains 12139 drug entries It records knowledge about two types of molecular networks: the chemical structure transformation network of drug development and the interaction network with metabolizing enzymes, target molecules, and other drugs [41].

Conclusion

In this chapter, we have discussed various used computational methods for DDI prediction. But first, we provided an overview of drug discovery and its main KG applications. The following chapter will present our novel method which is based on knowledge graphs embedding and attention neural networks.

Part II

Contributions

Chapter 3

Methods

Introduction

The prediction of drug-drug interactions (DDIs) plays an essential role in drug discovery. In recent years, deep learning methods have gained significant attention and have been extensively utilized for DDI prediction. Furthermore, leveraging knowledge graph (KG) embeddings as input to the predictive models has demonstrated improved performance and outcomes.

In this chapter, we are going to tackle the needed steps to build a new model named Attention KGDNN-DDI integrates knowledge graphs and attention neural networks. We will discuss our method in detail. The problem formulation is detailed in section 3.1. The DDI extraction and KG construction processes are presented in section 3.3. In sections 3.4 and 3.5 we respectively present our approaches for drug features learning and drug pair feature learning. Finally, the architecture of the Deep Neural Network used for model training and DDI prediction is presented in section 3.6.

3.1 Problem Formulation

We present the drug-drug interactions as a graph in which nodes indicate drugs and links indicate the potential interactions. Hence, in the given context, the problem is seen as a link prediction task. We consider two main aspects:

DDI matrix. We present a set N_d of drugs, and define the drug-drug interaction matrix Y $\in (0, 1)^{|N_d| \times |N_d|}$,. We note $|N_d|$ as the number of drugs. In the matrix, for every entry $y_{i,j} = 1$ $i, j \in N_d, j \neq i$ if its value is 1. This means that drug j interacts with drug i. We note that a value $y_{i,j} = 0$, does not necessarily mean the absence of interaction between drug pair (i, j). It signifies that the interaction may not have been found discovered yet in the KG.

Knowledge graph. Besides, the interactions between drug pairs, we note a directed knowledge graph $G = (V, E)$ where $e = (u, v) \in E$ represents an interaction between drugs u and v. Given the knowledge graph G and the DDI matrix Y, we aim to predict whether drug i ($i \in N_d$) has a potential interaction with drug j ($j \in N_d$, $j \neq i$) while such an interaction has not been observed before.

3.2 Our approach

Driven by the use of knowledge graphs (KG) in a number of research studies and particularly for DDI prediction, several works employed KG within supervised ML algorithms to extract drug features using different embedding methods [15]. On the other, the idea of incorporating attention mechanisms into deep neural networks has led to state-of-the-art results across a wide range of tasks [76]. However, attention models, which can utilize deep features, are mostly unexplored in the context of DDIs prediction. Therefore, we aimed to leverage the synergistic potential of knowledge graph embeddings and attention mechanisms by integrating them into our approach.

Figure 3.1 illustrates the overall architecture of our Attention KGDNN-DDI model. Our proposed method consists of four main components:

- DDI Extraction and KG Construction
- Drug features learning: It first converts the nodes in the graph into low-dimensional features.
- Drug-drug Pair Feature Learning: designs an attention neural network in order to learn drug-drug pairs' representations.
- DDI prediction: The DDI prediction is formulated as a binary classification based on a strategy of using the drug-drug pairs representation as input features.

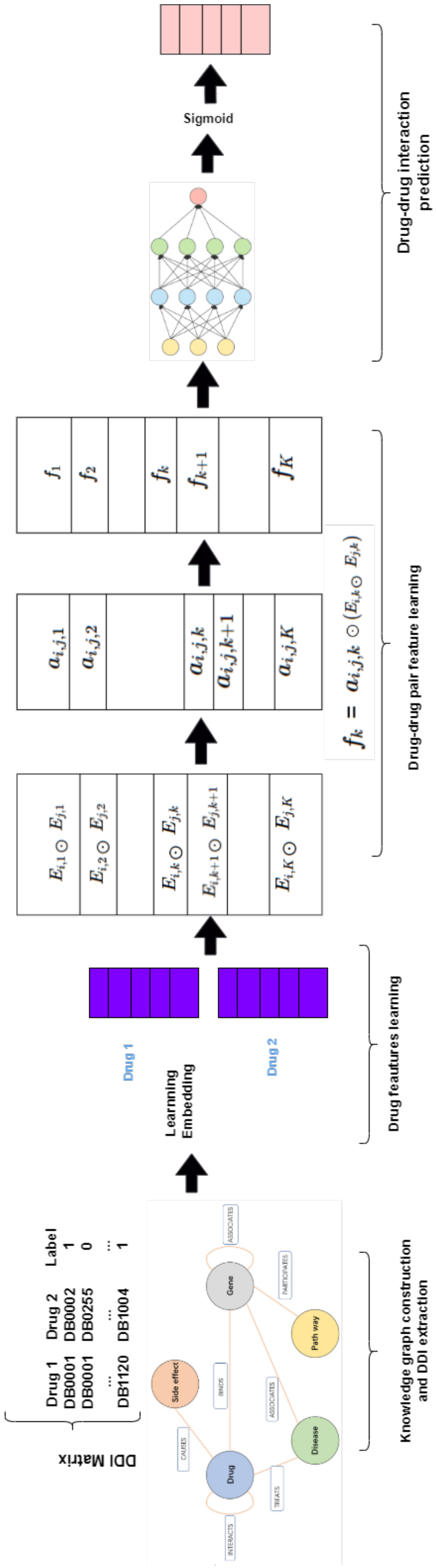


Figure 3.1: An overview of our proposed Attention KGDNN-DDI model.

3.3 DDI Extraction and KG Construction

DDI Extraction

For the DDI extraction, we first download the data from the public DrugBank data source, and then we parse the DDI information from the provided *xml* file. We then, compile an edge list of drug identifier combinations.

KG Construction.

We used a linked open biological dataset, called Bio2RDF, which interlinks data that contains multiple types of biological entities (e.g., drug, protein, and disease). The DrugBank dataset within the Bio2RDF project release 5.0 was used as the background knowledge graph. The DrugBank database combines detailed drug data (i.e. pharmacological, chemical, and pharmaceutical) with drug target (i.e. sequence, structure, and pathway) information. The extracted triples are formed in the form $(subject, predicate, object)$, indicating that the subject has a specified relation to the object.

We removed DDI links from the knowledge graph to eliminate bias on the prediction task. We exclude the information in the form of url: ddi-interactor-in and url:Drug-Drug-Interaction from DrugBank.

Hence, we note that two data sources from the pre-processing DrugBank dataset were used as input.

(i) the parsed DDI matrix that contains the drug-drug pairs.

(ii) The constructed knowledge graph.

3.4 Drug features learning

Drug features learning consists of generating the drug features using the information of our knowledge graph, as illustrated in figure 3.2. In order to utilize machine learning classifiers that typically require fixed-length vectors as input, we employed a knowledge graph (KG) embedding procedure to encode the information from the graph into dense vectors.

We used RDF2Vec as a knowledge graph embedding mechanism to generate drug features. The size of each drug feature vector is 200. It first generates sequences

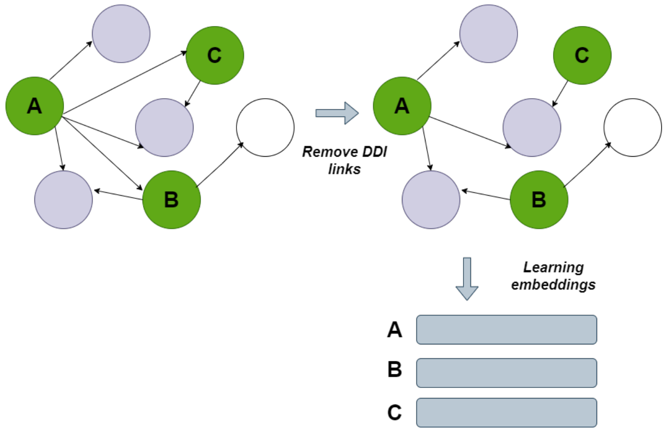


Figure 3.2: An example of the application of knowledge graph embedding to DDI prediction. A, B, and C nodes represent drug entities

from knowledge graphs by performing random walks, and next feeds those into the word2vec word embedding algorithm to compute vector representations for entities. We trained the word2vec model using CBOW architecture.

3.5 Drug-drug Pair Feature Learning

Our model is inspired by the successful applications of attention networks [18, 47]. The Drug-drug pair feature learning consists of an attention neural network to learn representations of drug-drug pairs by fusing learned embedding of drugs from drug feature networks.

First, we have as input the embedding vectors of individual drugs. We consider $[E_{ik}]$ and $[E_{jk}]$ the vector representation of drug i and drug j, respectively, and k denotes the dimensions of th vectors. Then, we need to learn the representation of every drug-drug pair, which is defined as follows :

$$F_{i,j} = a_{i,j} \odot (E_i \odot E_j) \tag{3.1}$$

where $\odot$ is the element-wise product and $a_{i,j} = (a_{i,j,1}, a_{i,j,2}, \ldots, a_{i,j,K})$ is a K-dimensional attention vector to capture the different importance of K dimensions in $(E_i \odot E_j)$. Precisely, $F_{i,j} = (f_1, f_2, \ldots, f_K)$ and $f_k = a_{i,j,k} \odot (E_{i,k} \odot E_{j,k})$ is the attention weight responsible for the *kth* dimesion, k = 1,2, … ,K.

Attention vector is employed to capture the importance of dimension k of the representation of a drug-drug pair, which is calculated as follows:

$$a_{i,j,k} = \frac{\exp(\hat{a}_{i,j,k})}{\sum_m \exp(a_{i,j,m})} \tag{3.2}$$

with $\hat{a}_{i,j} = \mathbf{V}^{\intercal}\mathrm{ReLU}(\mathbf{W}[E_i, E_j] + \mathbf{b})$, b is the bias vector, W the weight matrix, V^T the weight vector and and $\mathrm{ReLU}(x) = \max(0, x)$ is the activation function.

Consequently, the drug-drug pair feature learning component is capable to learn representations of drug-drug pairs taking into account different contributions of different features and their dimensions. Overall, the attention neural network will output the representations of drug-drug pairs to be used as input to the DDI prediction component.

3.6 Drug-drug interaction prediction

We treat the DDI prediction as a binary classification problem. Using the embeddings generated by the attention neural network, we can train now the deep neural network task, in which we aim to estimate the probability that a link with label 1 exists between two nodes.

First as shown in figure 3.3, we take the representations of drug-drug pairs as the input layer of the network. Next, the inputs will pass through multiple fully-connected hidden layers. We then applied the Rectified Linear Unit (RELU) as the activation function for all the hidden layers, each consisting of 32 units. Finally, the sigmoid function will assume output probabilities that indicate the likelihood of an instance belonging to only one particular class. We used the binary-cross entropy as the loss function.

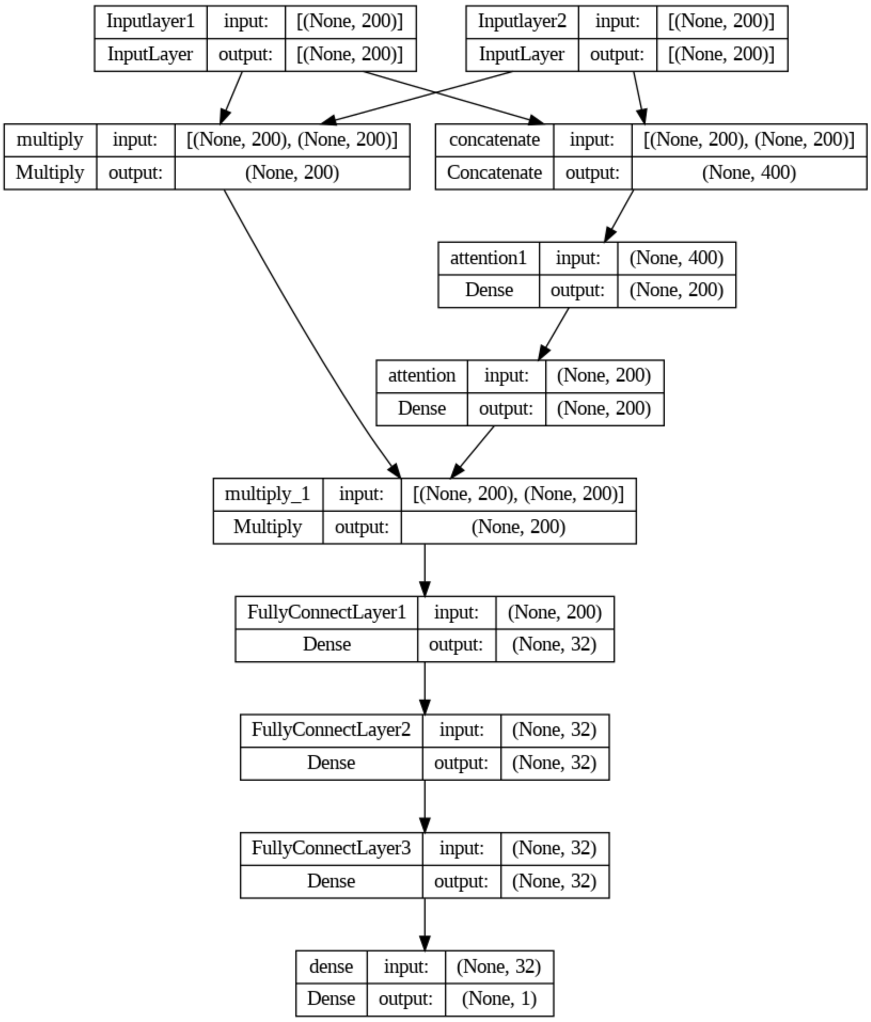

Inputlayer1 | input: | [(None, 200)]
InputLayer | output: | [(None, 200)]
Inputlayer2 | input: | [(None, 200)]
InputLayer | output: | [(None, 200)]
multiply | input: | [(None, 200), (None, 200)]
Multiply | output: | (None, 200)
concatenate | input: | [(None, 200), (None, 200)]
Concatenate | output: | (None, 400)
attention1 | input: | (None, 400)
Dense | output: | (None, 200)
attention | input: | (None, 200)
Dense | output: | (None, 200)
multiply_1 | input: | [(None, 200), (None, 200)]
Multiply | output: | (None, 200)
FullyConnectLayer1 | input: | (None, 200)
Dense | output: | (None, 32)
FullyConnectLayer2 | input: | (None, 32)
Dense | output: | (None, 32)
FullyConnectLayer3 | input: | (None, 32)
Dense | output: | (None, 32)
dense | input: | (None, 32)
Dense | output: | (None, 1)

Figure 3.3: Attention DNN Architecture

Conclusion

This chapter presents the key steps of our Attention KGDNN-DDI proposed method. We got through the different steps from KG construction to DDI prediction. In the following, the performance of our model will be evaluated in comparison to other ML methods.

Chapter 4

Experimental study

Introduction

In order to evaluate the prediction performance of the proposed method, we conducted a number of experiments. This chapter is organized as follows: we begin with representing the experimental setup in section 4.1. Next, we will detail the used evaluation criteria in section 4.2. In section 4.3 we will describe the baseline methods. Finally, the experimental results will be discussed in section 4.4.

4.1 Experimental setting and dataset

This section presents at first our experimental setting and the used dataset.

4.1.1 Experimental setting

In our experiment, Google Colab was used for model training, as it comes with free access to GPUs. The software stack consists of PyTorch, Scikit-learn, and Keras API with the TensorFlow backend.

We used RDF2Vec to generate drug features. For RDF2Vec input, we have generated walks with depth = 1,2,3,4, and walks per entity are equal to 250. Then, we trained the word2vec model using CBOW neural network architecture. The following parameters were used: window size = 5, number of iterations = 5, negative samples = 25, and dimension = 10.

Hyperparameter	Value
epochs	100
Batchsize	32
Learning rate	0.00001
Optimizer	RMSprop
Activation function	ReLU
Activation function (output)	Sigmoid

Table 4.1: Hyperparameters of our model

RMSprop with a learning rate of 0.00001 was chosen as an optimizer. Meanwhile, we set the number of epochs to 100, and other hyperparameters that were used in our model are listed in table 4.1.

To compare our model with other baselines, the dataset was randomly split into a training set, a validation set, and a test set, respectively, according to the ratio of 6:2:2.

4.1.2 Dataset

We evaluate our proposed model by using the DrugBank dataset. Dataset statistics are provided in table 4.2. As we need negative and positive for binary classification, the negative pairs were sampled from unknown pairs in sample size equivalent to the positive samples, in order to obtain a balanced dataset.

Dataset	Triples	Entities	Relation Types	Drugs	pairs
DrugBank	2,588,933	574,152	76	2551	288,856

Table 4.2: Dataset description

4.2 Evaluation Criteria

In our evaluation, to accurately measure the performance of our classifiers, we consider four different evaluation metrics which are accuracy, ROC-AUC, AUPR, and F1 score.

We note that TP and TN represent the number of drug-drug pairs with (without) interaction that were correctly identified respectively. Besides, FP and FN represent the number of drug-drug pairs with (without) interactions that were incorrectly predicted respectively.

Accuracy (ACC): Accuracy refers to the percentage of correct predictions of a model, regardless of the class (positive or negative). It is a useful metric when the classes are balanced. It can be expressed as:

$$Accuracy(ACC) = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

F1 score: The F1 score is a measure of a model's accuracy, taking into account both precision and recall. It ranges from 0 to 1, with 1 being the best possible score. The formula for the F1 score is:

$$F1score = 2 * \frac{(precision * recall)}{(precision + recall)} \tag{4.2}$$

where

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

and

$$Precision = \frac{TP}{TP + FP} \tag{4.4}$$

ROC-AUC: The Area Under the ROC Curve is a measure of a model's ability to differentiate between positive and negative classes. It measures the model's ability to correctly rank the probabilities of positive samples higher than negative samples. It ranges from 0 to 1, with 1 being the best possible score. The ROC curve is a plot of the true positive rate against the false positive rate. The AUC is the area under this curve.

AUPR: The AUPR (Area Under the Precision-Recall Curve) is another measure of a model's ability to distinguish between positive and negative classes, particularly in cases where the positive class is rare. The AUPR ranges from 0 to 1, with 1 being the best possible score. The precision-recall curve is a plot of precision against recall at different classification thresholds. The AUPR is the area under this curve.

4.3 Baselines

To validate the efficiency of our method, we compared the results of our method with different baseline models. We evaluated our method with various machine learning methods, including Logistic Regression and Naive Bayes.

Logistic regression:

Logistic regression applies the sigmoid function to produce a probability estimate for a label, making it a commonly employed method for binary classification tasks involving outcomes such as true or false, win or lose, or positive or negative. This probability output is then compared to a predetermined threshold, and the object is assigned a label based on the comparison.

Naive Bayes:

Naive Bayes is a probabilistic machine learning algorithm that is based on Bayes' theorem, which is used to calculate conditional probabilities based on prior knowledge. One of its key assumptions is that the features are conditionally independent of each other, given the class label.This is known as the ”naive” assumption and simplifies the calculation of probabilities by assuming that the occurrence of one feature is not influenced by the presence of another.

4.4 Experimental Results and Analysis

Table 4.3 below illustrated the obtained experimental results of the performance comparison between our method and baseline models on the DrugBank dataset. The best results are designed with bold values. The overall performances are measured by four evaluation metrics, accuracy, AUC-ROC, AUPR, and F1-score.In general, we could outperform the baseline classifiers, consistently with our proposed architecture. We obtained up to 0.9992, 0.9997, 0.9929, and 0.9982 for Accuracy, ROC-AUC, AUPR, and F-score respectively.

Methods	LR	NB	Our model
ACC	0.8064	0.6836	**0.9992**
ROC-AUC	0.8787	0.7145	**0.9997**
AUPR	0.8772	0.6836	**0.9996**
F-score	0.8086	0.6312	**0.9982**

Table 4.3: Performance Metrics Comparison: The embedding generated by RDF2Vec was used in the experiments.

The main objective of modeling is to find the minimum value loss at each iteration. The convergence needs to happen with a decreasing trend. It happens when the errors produced by the model in training gets to a minimum. After the 20th epoch, we can see in figure 4.1 that the training of the model starts to converge and the errors are lower, decremental, and within a smaller range.

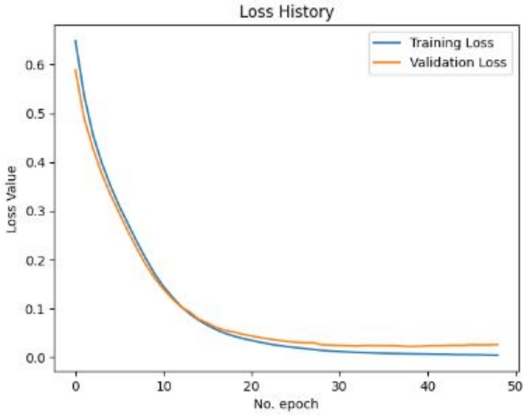


Figure 4.1: Loss curve during training

As we observe in figure 4.2, the validation accuracy follows the training accuracy with being slightly inferior. Thus, the model is able to predict correctly instances that are not part of the training set.

Confusion Matrix is a table that is used for evaluating the performance of a classification model by comparing the predicted and actual values for a set of test data. As illustrated in figure 4.3, out of 101380 predictions only 14894 were miss-classified.

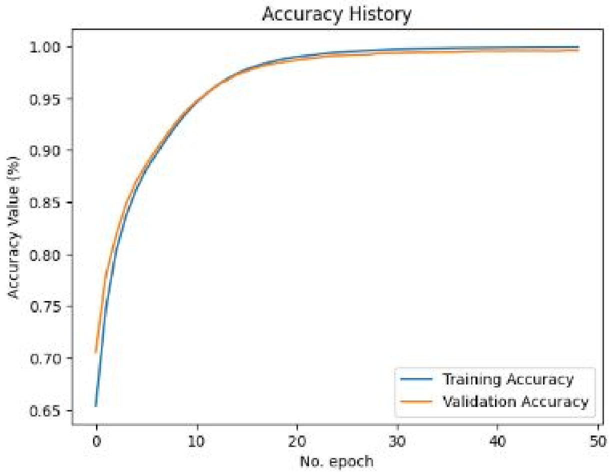


Figure 4.2: Accuracy curve during training

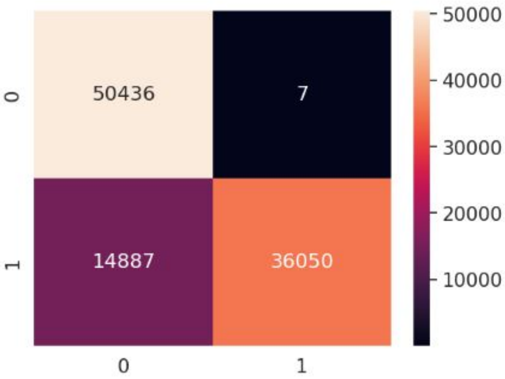


Figure 4.3: Confusion Matrix of our proposed model

Conclusion

This chapter represented the evaluation of our model based on different metrics. We demonstrated that Attention KGDNN-DDI was able to perform better than other ML methods.

Conclusion

The main focus of this thesis is on the prediction of drug-drug interactions using knowledge graphs and attention neural networks. We recognized the severity of adverse drug reactions, thus we emphasize the need for predictive methods.

To address this challenge, we propose the integration of drug-related knowledge graphs. However, the format of the data within knowledge graphs is not directly compatible with typical classifiers. Therefore, we employed knowledge graph embedding techniques to transform the data into dense vector representations. These representations are then utilized as inputs for the drug-drug pairs features component using attention. Then the generated embeddings are fed into the DNN network. The overall objective of the work is to harness the power of knowledge graphs to integrate drug-related information and leverage advanced deep-learning and attention-based techniques to predict drug-drug interactions.

One limitation of our approach is the inability to come up with explanations for the predicted DDIs. As future research directions, we would like to invest more in explainable AI models, as our model does not provide the justification underlying each prediction. It would also be interesting to explore multi-type feature fusing by concatenating structural drug features and topological features extracted from KGs.

Bibliography

[1] A. Bordes, N. Usunier, A. G.-D. J. W., and Yakhnenko., O. Translating embeddings for modeling multi-relational data. *In Proc. of NIPS* (2013).

[2] A. Hogan, E. Blomqvist, M. C.-C. D. G. D. M. C. G. S. K. J. E. L. G. R. N. S. N. A.-C. A. P. S. M. R. A. R. L. S. J. S. S. S. A. Z. Knowledge graphs. *ACM Computing Surveys 54*, 4 (may 2022). [Accessed 30 March. 2023].

[3] Abbas, K., Abbasi, A., Dong, S., Niu, L., Yu, L., Chen, B., Cai, S.-M., and Hasan, Q. Application of network link prediction in drug discovery. *BMC bioinformatics 22* (2021), 1–21.

[4] Abdelaziz, I., Fokoue, A., Hassanzadeh, O., Zhang, P., and Sadoghi, M. Large-scale structural and textual similarity-based mining of knowledge graph to predict drug–drug interactions. *Journal of Web Semantics 44* (2017), 104–117.

[5] Alshahrani, M., Khan, M. A., Maddouri, O., Kinjo, A. R., Queralt-Rosinach, N., and Hoehndorf, R. Neuro-symbolic representation learning on biological knowledge graphs. *Bioinformatics 33*, 17 (2017), 2723–2730.

[6] Amador-Domínguez, E., Serrano, E., Manrique, D., and De Paz, J. F. Prediction and decision-making in intelligent environments supported by knowledge graphs, a systematic review. *Sensors 19*, 8 (2019), 1774.

[7] Arbel, N.

[8] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. Dbpedia: A nucleus for a web of open data. In *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference,*

ISWC 2007+ ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings (2007), Springer, pp. 722–735.

[9] B. Yang, W. Yih, X. H.-J. G., and Deng., L. Embedding entities and relations for learning and inference in knowledge bases. *In Proc. of ICLR* (2015).

[10] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[11] Bebis, G., and Georgiopoulos, M. Feed-forward neural networks. *Ieee Potentials 13*, 4 (1994), 27–31.

[12] Bollacker, K., Cook, R., and Tufts, P. Freebase: A shared database of structured general human knowledge. In *AAAI* (2007), vol. 7, pp. 1962–1963.

[13] Bonner, S., Barrett, I. P., Ye, C., Swiers, R., Engkvist, O., Bender, A., Hoyt, C. T., and Hamilton, W. L. A review of biomedical datasets relating to drug discovery: a knowledge graph perspective. *Briefings in Bioinformatics 23*, 6 (2022).

[14] Bougiatiotis, K., Aisopos, F., Nentidis, A., Krithara, A., and Paliouras, G. Drug-drug interaction prediction on a biomedical literature knowledge graph. 122–132.

[15] Celebi, R., Uyar, H., Yasar, E., Gumus, O., Dikenelli, O., and Dumontier, M. Evaluation of knowledge graph embedding approaches for drug-drug interaction prediction in realistic settings. *BMC bioinformatics 20*, 1 (2019), 1–14.

[16] Chauhan, R., Ghanshala, K. K., and Joshi, R. Convolutional neural network (cnn) for image detection and recognition. In *2018 first international conference on secure cyber computing and communication (ICSCCC)* (2018), IEEE, pp. 278–282.

[17] Chen, F., Wang, Y.-C., Wang, B., and Kuo, C.-C. J. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing 9* (2020), e15.

[18] Cheng, Z., Ding, Y., He, X., Zhu, L., and Song, X. Aˆ 3ncf: An adaptive aspect attention model for rating prediction.

[19] Consortium, G. O. The gene ontology (go) database and informatics resource. *Nucleic acids research 32*, suppl_1 (2004), D258–D261.

[20] Consortium, U. Uniprot: a hub for protein information. *Nucleic acids research 43*, D1 (2015), D204–D212.

[21] da Silva Rocha, S. F., Olanda, C. G., Fokoue, H. H., and Sant'Anna, C. M. Virtual screening techniques in drug discovery: review and recent applications. *Current topics in medicinal chemistry 19*, 19 (2019), 1751–1767.

[22] Deepika, S., and Geetha, T. A meta-learning framework using representation learning to predict drug-drug interaction. *Journal of biomedical informatics 84* (2018), 136–147.

[23] Deore, A. B., Dhumane, J. R., Wagh, R., and Sonawane, R. The stages of drug discovery and development process. *Asian Journal of Pharmaceutical Research and Development 7*, 6 (2019), 62–67.

[24] Dhillon, A., and Verma, G. K. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence 9*, 2 (2020), 85–112.

[25] Duelen, R., Corvelyn, M., Tortorella, I., Leonardi, L., Chai, Y. C., and Sampaolesi, M. Medicinal biotechnology for disease modeling, clinical therapy, and drug discovery and development. *Introduction to Biotech Entrepreneurship: From Idea to Business: A European Perspective* (2019), 89–128.

[26] Endris, K. M., Galkin, M., Lytra, I., Mami, M. N., Vidal, M.-E., and Auer, S. Querying interlinked data by bridging rdf molecule templates. *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXIX: Special Issue on Database-and Expert-Systems Applications* (2018), 1–42.

[27] Euler, L. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae* (1741), 128–140.

[28] F. Manola, E. M., and McBride, B. Rdf 1.1 primer. *W3C* (June 2014). [Accessed 30 March. 2023].

[29] Ferdousi, R., Safdari, R., and Omidi, Y. Computational prediction of drug-drug interactions based on drugs functional similarities. *Journal of biomedical informatics 70* (2017), 54–64.

[30] Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 2414–2423.

[31] Gottlieb, A., Stein, G. Y., Ruppin, E., and Sharan, R. Predict: a method for inferring novel drug indications with application to personalized medicine. *Molecular systems biology 7*, 1 (2011), 496.

[32] Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics 37*, 3 (2020), 362–386.

[33] Han, K., Cao, P., Wang, Y., Xie, F., Ma, J., Yu, M., Wang, J., Xu, Y., Zhang, Y., and Wan, J. A review of approaches for predicting drug–drug interactions based on machine learning. *Frontiers in Pharmacology 12* (2022), 3966.

[34] He, Y., Shen, Z., Zhang, Q., Wang, S., and Huang, D.-S. A survey on deep learning in dna/rna motif mining. *Briefings in Bioinformatics 22*, 4 (2021), bbaa229.

[35] Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6*, 02 (1998), 107–116.

[36] Hopfield, J. J. ”neural networks and physical systems with emergent collective computational abilities”. *Proceedings of the National Academy of Sciences* (1980).

[37] Hubel, D., and Wiesel, T. Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology 195* (1968).

[38] Joshi, P., Masilamani, V., and Mukherjee, A. A knowledge graph embedding-based approach to predict the adverse drug reactions using a deep neural network. *Journal of Biomedical Informatics 132* (2022), 104122.

[39] Jupp, S., Malone, J., Bolleman, J., Brandizi, M., Davies, M., Garcia, L., Gaulton, A., Gehant, S., Laibe, C., Redaschi, N., et al. The ebi rdf platform: linked open data for the life sciences. *Bioinformatics 30*, 9 (2014), 1338–1339.

[40] Kamath, U., Liu, J., and Whitaker, J. *Deep learning for NLP and speech recognition*, vol. 84. Springer, 2019.

[41] Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y., and Morishima, K. Kegg: new perspectives on genomes, pathways, diseases and drugs. *Nucleic acids research 45*, D1 (2017), D353–D361.

[42] Karim, M. R., Cochez, M., Jares, J. B., Uddin, M., Beyan, O., and Decker, S. Drug-drug interaction prediction based on knowledge graph embeddings and convolutional-lstm network. 113–123.

[43] Li, Y. Research and analysis of semantic search technology based on knowledge graph. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)* (2017), vol. 1, IEEE, pp. 887–890.

[44] Lin, X., Quan, Z., Wang, Z.-J., Ma, T., and Zeng, X. Kgnn: Knowledge graph neural network for drug-drug interaction prediction. 2739–2745.

[45] Liu, G., and Guo, J. Bidirectional lstm with attention mechanism and convolutional layer for text classification. *Neurocomputing 337* (2019), 325–338.

[46] Liu, S., Zhang, Y., Cui, Y., Qiu, Y., Deng, Y., Zhang, Z., and Zhang, W. Enhancing drug-drug interaction prediction using deep attention neural networks. *IEEE/ACM transactions on computational biology and Bioinformatics 20*, 2 (2022), 976–985.

[47] Liu, S., Zhang, Y., Cui, Y., Qiu, Y., Deng, Y., Zhang, Z., and Zhang, W. Enhancing drug-drug interaction prediction using deep attention neural networks. *IEEE/ACM transactions on computational biology and Bioinformatics 20*, 2 (2022), 976–985.

[48] M. Wang, L. Q., and Wang, X. A survey on knowledge graph embeddings for link prediction. *Symmetry 13*, 3 (may 2021). [Accessed 30 March. 2023].

[49] McCulloch, W. S., and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics 5* (1943), 115–133.

[50] Mohamed, S. K., Nounu, A., and Nováček, V. Drug target discovery using knowledge graph embeddings. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing* (2019), pp. 11–18.

[51] Mohamed, S. K., Nounu, A., and Nováček, V. Biological applications of knowledge graph embedding models. *Briefings in bioinformatics 22*, 2 (2021), 1679–1693.

[52] Nickel, M., and Tresp., V. Tensor factorization for multi-relational learning. *In Proc. of ECML-PKDD.* (2013).

[53] Niu, Z., Zhong, G., and Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing 452* (2021), 48–62.

[54] Nolin, M.-A., Ansell, P., Belleau, F., Idehen, K., Rigault, P., Tourigny, N., Roe, P., Hogan, J. M., and Dumontier, M. Bio2rdf network of linked data. In *Semantic Web Challenge; International Semantic Web Conference (ISWC 2008)* (2008), Citeseer.

[55] Olayan, R. S., Ashoor, H., and Bajic, V. B. Ddr: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. *Bioinformatics 34*, 7 (2018), 1164–1173.

[56] Onakpoya, I. J., Heneghan, C. J., and Aronson, J. K. Post-marketing withdrawal of 462 medicinal products because of adverse drug reactions: a systematic review of the world literature. *BMC medicine 14*, 1 (2016), 1–11.

[57] Pak, M., and Kim, S. A review of deep learning in image recognition. In *2017 4th international conference on computer applications and information processing technology (CAIPT)* (2017), IEEE, pp. 1–3.

[58] Phung, V. H., and Rhee, E. J. A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. *Applied Sciences 9*, 21 (2019), 4500.

[59] Pla Karidi, D., Stavrakas, Y., and Vassiliou, Y. Tweet and followee personalized recommendations based on knowledge graphs. *Journal of Ambient Intelligence and Humanized Computing 9*, 6 (2018), 2035–2049.

[60] Raman, K. Construction and analysis of protein–protein interaction networks. *Automated experimentation 2* (2010), 1–11.

[61] Reinanda, R., Meij, E., de Rijke, M., et al. Knowledge graphs: An information retrieval perspective. *Foundations and Trends® in Information Retrieval 14*, 4 (2020), 289–444.

[62] Ristoski, P., and Paulheim, H. Rdf2vec: Rdf graph embeddings for data mining. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I 15* (2016), Springer, pp. 498–514.

[63] Rohani, N., and Eslahchi, C. Drug-drug interaction predicting by neural network using integrated similarity. *Scientific reports 9*, 1 (2019), 13645.

[64] Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review 65*, 6 (1958), 386.

[65] Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S., and Sontag, D. Learning a health knowledge graph from electronic medical records. *Scientific reports 7*, 1 (2017), 1–11.

[66] Ryu, J. Y., Kim, H. U., and Lee, S. Y. Deep learning improves prediction of drug–drug and drug–food interactions. *Proceedings of the National Academy of Sciences 115*, 18 (2018), E4304–E4311.

[67] Scannell, J. W., Blanckley, A., Boldon, H., and Warrington, B. Diagnosing the decline in pharmaceutical r&d efficiency. *Nature reviews Drug discovery 11*, 3 (2012), 191–200.

[68] Schmidhuber, S. H. J. ”long short-term memory”. *Neuaral Computation* (1997).

[69] Schneider, E. W. Course modularization applied: The interface system and its implications for sequence control and data analysis.

[70] Schroedl, S. Machine learning for drug discovery in a nutshell - part i. *Medium* (2018).

[71] Segura-Bedmar, I., Martínez Fernández, P., and Herrero Zazo, M. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). Association for Computational Linguistics.

[72] Segura-Bedmar, I., Martínez Fernández, P., and Sánchez Cisneros, D. The 1st ddiextraction-2011 challenge task: Extraction of drug-drug interactions from biomedical texts.

[73] Sharma, J.

[74] Shewalkar, A., Nyavanandi, D., and Ludwig, S. A. Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru. *Journal of Artificial Intelligence and Soft Computing Research 9*, 4 (2019), 235–245.

[75] Shtar, G., Rokach, L., and Shapira, B. Detecting drug-drug interactions using artificial neural networks and classic graph similarity measures. *PloS one 14*, 8 (2019), e0219796.

[76] Soydaner, D. Attention mechanism in neural networks: where it comes and where it goes. *Neural Computing and Applications 34*, 16 (2022), 13371–13385.

[77] Suchanek, F. M., Kasneci, G., and Weikum, G. Yago: A large ontology from wikipedia and wordnet. *Journal of Web Semantics 6*, 3 (2008), 203–217.

[78] Sun, X., Ma, L., Du, X., Feng, J., and Dong, K. Deep convolution neural networks for drug-drug interaction extraction. 1662–1668.

[79] T. Trouillon, J. Welbl, S. R.- G., and Bouchard, G. Complex embeddings for simple link prediction. *In Proc. of ICML* (2016).

[80] Tarwani, K. M., and Edem, S. Survey on recurrent neural network in natural language processing. *Int. J. Eng. Trends Technol 48*, 6 (2017), 301–304.

[81] Tatonetti, N. P., Ye, P. P., Daneshjou, R., and Altman, R. B. Data-driven prediction of drug effects and interactions. *Science translational medicine 4*, 125 (2012), 125ra31–125ra31.

[82] Thorn, C. F., Klein, T. E., and Altman, R. B. Pharmgkb: the pharmacogenomics knowledge base. *Pharmacogenomics: Methods and Protocols* (2013), 311–320.

[83] Vamathevan, J., Clark, D., Czodrowski, P., Dunham, I., Ferran, E., Lee, G., Li, B., Madabhushi, A., Shah, P., Spitzer, M., et al. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery 18*, 6 (2019), 463–477.

[84] Vo, T. H., Nguyen, N. T. K., Kha, Q. H., and Le, N. Q. K. On the road to explainable ai in drug-drug interactions prediction: A systematic review. *Computational and Structural Biotechnology Journal* (2022).

[85] Vo, T. H., Nguyen, N. T. K., and Le, N. Q. K. Improved prediction of drug-drug interactions using ensemble deep neural networks. *Medicine in Drug Discovery 17* (2023), 100149.

[86] Vrandečić, D., and Krötzsch, M. Wikidata: a free collaborative knowledgebase. *Communications of the ACM 57*, 10 (2014), 78–85.

[87] Wang, S., Hu, L., Cao, L., Huang, X., Lian, D., and Liu, W. Attention-based transactional context embedding for next-item recommendation. In *Proceedings of the AAAI conference on artificial intelligence* (2018), vol. 32.

[88] Wishart, D. S., Knox, C., Guo, A. C., Cheng, D., Shrivastava, S., Tzur, D., Gautam, B., and Hassanali, M. Drugbank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic acids research 36*, suppl_1 (2008), D901–D906.

[89] Xiong, G., Chen, X., Shuo, N., Lv, Y., Zhu, F., Qu, T., and Ye, P. Collaborative optimization of cyber physical social systems for urban transportation based on knowledge automation. *IFAC-PapersOnLine 53*, 5 (2020), 572–577.

[90] Yani, M., and Krisnadhi, A. A. Challenges, techniques, and trends of simple knowledge graph question answering: a survey. *Information 12*, 7 (2021), 271.

[91] Yoo, S., and Jeong, O. An intelligent chatbot utilizing bert model and knowledge graph. *Journal of Society for e-Business Studies 24*, 3 (2020).

[92] Yue, X., Wang, Z., Huang, J., Parthasarathy, S., Moosavinasab, S., Huang, Y., Lin, S. M., Zhang, W., Zhang, P., and Sun, H. Graph embedding on biomedical networks: methods, applications and evaluations. *Bioinformatics 36*, 4 (2020), 1241–1251.

[93] Z. Sun, Z. Deng, J. N., and Tang., J. . rotate: Knowledge graph embedding by relational rotation in complex space. *In Proc. of ICLR* (2019).

[94] Zamir, A. R., Wu, T.-L., Sun, L., Shen, W. B., Shi, B. E., Malik, J., and Savarese, S. Feedback networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 1308–1317.

[95] Zhang, C., Lu, Y., and Zang, T. Cnn-ddi: a learning-based method for predicting drug–drug interactions using convolution neural networks. *BMC bioinformatics 23*, 1 (2022), 1–12.

[96] Zhang, S., Yu, C., and Xu, C. Integrating knowledge graph and bi-lstm for drug-drug interaction predication. 763–771.

[97] Zitnik, M., Agrawal, M., and Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics 34*, 13 (2018), i457–i466.