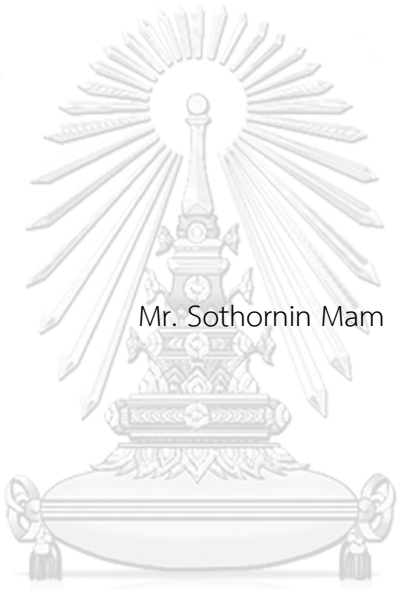
LINK PREDICTION USING DEEP LEARNING APPROACH FOR TYPE 2 DIABETES DRUG REPURPOSING

Mr. Sothornin Mam



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2022

การทำนายความเชื่อมโยงโดยวิธีการเรียนรู้เชิงลึกสำหรับการนำยามาหาข้อบ่งใช้ในโรคเบาหวานชนิดที่ 2

นายสุธรณินทร์ มอม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2565

Thesis Title	LINK PREDICTION USING DEEP LEARNING APPROACH FOR TYPE 2 DIABETES DRUG REPURPOSING
By	Mr. Sothornin Mam
Field of Study	Computer Science
Thesis Advisor	Associate Professor PEERAPON VATEEKUL, Ph.D.
Thesis Co Advisor	Associate Professor DUANGDAO WICHADAKUL, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Science

..... Dean of the FACULTY OF ENGINEERING

(Professor SUPOT TEACHAVORASINSKUN, D.Eng.)

THESIS COMMITTEE

..... Chairman

(Professor BOONSERM KIJSIRIKUL, D.Eng.)

..... Thesis Advisor

(Associate Professor PEERAPON VATEEKUL, Ph.D.)

..... Thesis Co-Advisor

(Associate Professor DUANGDAO WICHADAKUL, Ph.D.)

..... Examiner

(PUNNARAI SIRICHAROEN, Ph.D.)

..... External Examiner

(Thanapat Kangkachit, Ph.D.)

สุธรณินทร์ มอม : การทำนายความเชื่อมโยงโดยวิธีการเรียนรู้เชิงลึกสำหรับการนำยามาหาข้อบ่งใช้ในโรคเบาหวานชนิดที่ 2. (LINK PREDICTION USING DEEP LEARNING APPROACH FOR TYPE 2 DIABETES DRUG REPURPOSING) อ.ที่ปรึกษาหลัก : รศ. ดร.พีรพล เวทีกูล, อ.ที่ปรึกษาร่วม : รศ. ดร.ดวงดาว วิชาดากุล

โรคเบาหวานชนิดที่ 2 เป็นโรคเรื้อรังที่เกิดขึ้นมาอย่างยาวนานและปัจจุบันยังไม่มีการค้นพบการรักษาผู้ป่วยโรคดังกล่าวให้หายขาดได้ การนำยามาหาข้อบ่งใช้ใหม่ (drug repurposing) จากคลังยาที่ใช้ในการรักษาโรคอื่น ๆ จึงเป็นวิธีการหนึ่งที่มีความสำคัญในการรักษาโรคเบาหวานชนิดที่ 2 จากการค้นคว้าที่ผ่านมาพบว่ามีการนำวิธีการเรียนรู้เชิงลึก (deep learning) มาประยุกต์ใช้โดยใช้การทำนายความเชื่อมโยง (link prediction) จากตัวแทนข้อมูลกราฟ (graph representation) ทั้งโครงสร้างของกราฟและข้อความแยกกัน เป็นเหตุให้สมรรถนะของโมเดลค่อนข้างจำกัด งานวิจัยนี้จึงนำเสนอวิธีการใหม่ในการนำโมเดลที่ได้จากการเรียนรู้เชิงลึกมาพยากรณ์ความเชื่อมโยงระหว่างยากับการรักษาโรคเบาหวานชนิดที่ 2 โดยโมเดลใหม่นี้พัฒนาขึ้นจากโมเดล transformer ที่เป็นโมเดลการเรียนรู้เชิงลึกใหม่ที่มีประสิทธิภาพมากขึ้น การพัฒนาโมเดลในงานวิจัยนี้เป็นการฝังตัวแทนข้อมูลกราฟจาก (1) โครงสร้างของกราฟฝังจากข้อมูลการเชื่อมโยงระหว่างโหนดกับโหนดรอบข้าง และ (2) ข้อมูลเชิงความหมายที่สกัดจากชื่อและคำอธิบายของโหนด โดยทำการทดลองบนข้อมูลของโรคเบาหวานชนิดที่ 2 ที่เรียกค้นจากฐานข้อมูล PubMed และ UMLS Metathesaurus พบว่าผลลัพธ์ของโมเดลใหม่นี้ที่นำตัวแทนข้อมูลกราฟทั้งสองมาวิเคราะห์นั้นมีค่าประสิทธิภาพสูงกว่าจากโมเดลดั้งเดิมร้อยละ 77.17 ตามระดับคะแนน mean reciprocal rank โดยการวัดความสามารถในการค้นพบยา (drug discovery) กล่าวคือมีประสิทธิภาพสูงกว่าโมเดลที่ใช้ตัวแทนข้อมูลเพียงประเภทเดียว เช่น StAR หรือ HittER เมื่อนำโมเดลที่ได้มาจำแนกหาข้อบ่งใช้ใหม่ของยาในการรักษาโรคเบาหวาน พบว่ารายการยาที่ได้จากโมเดลนี้มีความเหมาะสมในการรักษาโรคเบาหวานชนิดที่ 2

สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์	ลายมือชื่อนิสิต .....
ปีการศึกษา	2565	ลายมือชื่อ อ.ที่ปรึกษาหลัก .....
		ลายมือชื่อ อ.ที่ปรึกษาร่วม .....

6470163721 : MAJOR COMPUTER SCIENCE
KEYWORD: Deep learning, Type 2 diabetes, Link prediction, Drug repurposing

Sothornin Mam : LINK PREDICTION USING DEEP LEARNING APPROACH FOR TYPE 2 DIABETES DRUG REPURPOSING. Advisor: Assoc. Prof. PEERAPON VATEEKUL, Ph.D. Co-advisor: Assoc. Prof. DUANGDAO WICHADAKUL, Ph.D.

There is still no effective treatment for type 2 diabetes, which has been on the rise for years. By repositioning current medications for new indications, drug repurposing can aid in the discovery of novel medications. Deep learning has recently been applied to this problem via link prediction utilizing a graph representation that learns from either the structure of a graph or the semantic meaning of entity text. However, because they used a single representation as the basis for their work without making any model improvements, earlier attempts still had restricted performance. In this study, we suggest a new deep-learning approach for the drug repurposing of entities associated with type 2 diabetes. Transformer, a current deep learning network, serves as the foundation of our model's architecture. Regarding our link prediction in the graph, each entity is embedded utilizing both (1) structural information embedded from the node and its neighbor nodes and (2) semantic information retrieved from its name and descriptions. The experiment was conducted using type 2 diabetes data gathered from PubMed and UMLS Metathesaurus. The findings demonstrated that our combined model can outperform other models that only contain a single module, i.e. StAR and HittER, by exhibiting an increase of 77.17% on the mean reciprocal rank score for the drug discovery task. Finally, using the model for drug repurposing, we can identify several medications that may be employed to treat type 2 diabetes.

Field of Study:	Computer Science	Student's Signature .....
Academic Year:	2022	Advisor's Signature .....
		Co-advisor's Signature .....

ACKNOWLEDGEMENTS

I am deeply grateful to my advisor, Associate Professor Peerapon Vateekul, for providing consistent advice and unwavering support throughout my research. His guidance and encouragement were instrumental in successfully completing and publishing this study. I am truly fortunate to have pursued my master's degree under his mentorship, as his patience and attentiveness during challenging moments have been invaluable. I would also like to express my heartfelt appreciation to my co-advisor, Associate Professor Duangdao Wichadakul. Her expertise in biomedical sciences greatly influenced the entire research process. As the program head, she was one of the first professors I approached for guidance. I am grateful to her for selecting me for the master's program and consistently providing reassurance that I am deserving of my place here. Furthermore, I extend my deepest gratitude to all my lab mates, particularly Passin Pornvoraphat and Passakron Phuangthongkham, for being an exceptional team and consistently offering their assistance whenever I needed it. In addition, I would like to express my heartfelt appreciation to my two Ph.D. seniors, Manassakan Sanayha and Phattharat Songthung, for their constant positive encouragement and support throughout my graduate studies. Lastly, I am incredibly grateful to my entire family for their unwavering belief in me and their endless love and support. During my study, I am a student under Scholarship for International Graduate Students supported by Graduate School, Chulalongkorn University. This research project is supported by the 72nd Anniversary of His Majesty King Bhumibol Adulyadej Scholarship and the 90th Anniversary Chulalongkorn University Fund (Ratchadapiseksomphot Endowment Fund).

Sothornin Mam

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Diabetes is a chronic disease related to the abnormality of the pancreas functionality. Diabetes can be divided into two different types: type 1 diabetes or insulin-dependent, and type 2 diabetes or non-insulin-dependent. Both types are similar since all are related to the problem posed by the way our body uses insulin. While type 1 diabetes patients produce less insulin which as a result is not enough for body usage, type 2 diabetes is concerned with how our body inefficiently uses insulin. It is reported that greater than 95% of the people who are diagnosed with diabetes are type 2 diabetes which also shows symptoms later if we compare it to type 1 diabetes [1]. There is no known cure for diabetes yet. As for the first type the patient needs to inject insulin regularly. Most drugs used by type 2 diabetes patients are mainly consumed to help the body improve the usage of insulin as well as control and reduce the amount of insulin when it is produced excessively [2].

Although there is no way of completely recovering from diabetes, some drugs are recommended to help lessen the condition. There are many types of drugs recommended for coping with diabetes, however, there is also the possibility of using a cure for other diseases for treating diabetes as well. Drug repurposing is the method of looking for new usage of clinical drugs. These drugs which were not known to be able to be used for a certain disease would go through certain procedures so the researcher can be sure whether the new indication would work on another disease. Some popular approach includes molecular-docking-based, and drug screening which involves looking at how the drug affects the host protein of the disease [3].

Researchers have been publishing academic works on biomedical for ages. This knowledge is very useful for our understanding of the nature of different types of diseases and ways to cope with them which provides an imminent source of data

for lots of useful purposes. However, text from academic papers is largely unstructured and cannot be immediately useful to be shown as insight. But thanks to the development of natural language processing (NLP), which mainly concerns text data, has eased the way we deal with all the text information in a way that we could not before. All the long unstructured text from academic research can then be processed to extract useful knowledge by using what is known as name entity recognition and relation extraction. With entity recognition, real-world entities such as people, places, or biomedical fields such as proteins, drugs, or diseases are extracted from sentences and classified with each of their types. As the entities appear in the sentences, relation extraction would infer the type of semantic relation that different entities appear in the same sentence or even cross-sentences. Through these two tools, multi-relation sets are combined into a network of entities which different relations connecting them all into a knowledge graph. However, knowledge graphs can be incomplete. A task that is concerned in the knowledge graph field is the knowledge graph complete or specifically link prediction. The goal of the link prediction task is to predict whether there is a new possible link between the entity existing in the graph. Hence, through the knowledge graph of biomedical entities, we can infer a new connection between drugs and disease which is not known to be related before.

The real-world names of most knowledge graph entities and relations can be used to represent natural language text textually. Many pre-trained language models are capable of providing information from their textual meaning in human language by developing a language model paired with transformer architecture [4]. BERT [5] is an instance of a transformer-based language model that only has an encoder component. A model like this is frequently used to extract textual embedding and may be used for many downstream applications. Since a pre-trained model is trained on an extensive text corpus, such as the text of Wikipedia, it is a valuable source for textual encoding. As a result, the textual embedding produced by the language model can be considered a very potent feature used in training the link prediction model. The authors of [6] proposed the KG-BERT model, which applies a language

model to knowledge graph tasks. The text includes representations of both entities and relations that are organized in a series. KG-BERT likewise employs a single transformer model but experiences severe overload due to the need to compute all conceivable combinations of entities and relations during the inference stage, which slows down the execution time. To address the issue encountered by KG-BERT, the "Structure-augmented text representation" (StAR) model has been implemented [7]. Additionally, while entities and relations can be considered textual data in language models, a different type of transformer model, namely HittER, views each entity and relation as a single token [8]. Instead of depending on the semantic information offered by language models, these tokens are subsequently put into the model to learn how nodes interact with their surrounding nodes.

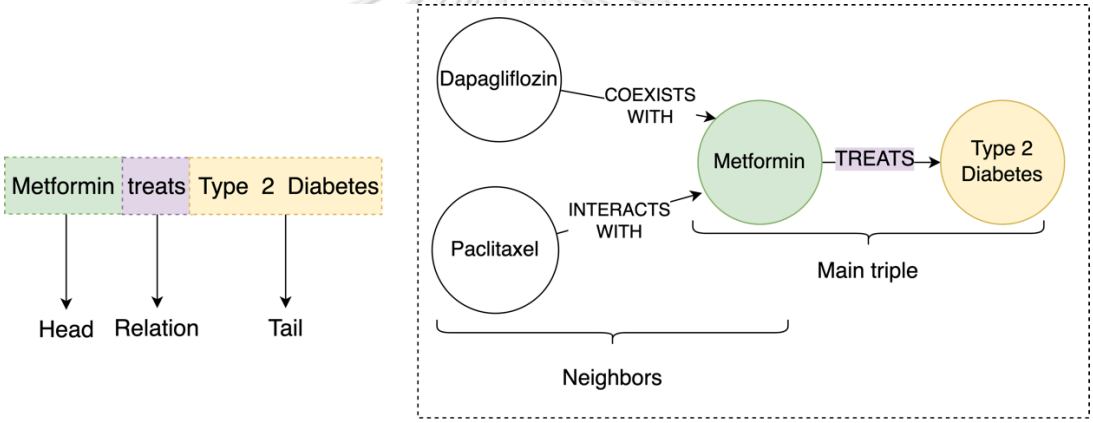


Figure 1 Comparing the two methods of representing a graph: (a) Textual representation, which consists of a head, relation, and tail sequence of text; and (b) Structural representation, which shows the primary triple in connection to its neighbors.

In Figure 1, two separate transformer-based models, StAR and HittER, embed knowledge graphs using various forms of data. A series of texts serve as an instance of textual representation. For example, the phrase "Metformin treats Type 2 Diabetes" is the textual representation of the entities "Metformin" and "Type 2 Diabetes", which are linked by the relation "TREATS". Regarding structural representation, the model instead learns embedding from surrounding nodes with

various kinds of relations rather than using the semantic meaning of nodes. Additional insights into the link prediction model can be given when textual and structural representations are combined. Although semantic or textual information tasks have previously proven to be highly effective, the relevance of graph structure information, that stores a node's relationship with its neighbor nodes, cannot be ignored. We assume that both representations work well together and can improve link prediction task performance.

In this work, we will focus on how we could use link prediction task to propose new drugs as the treatment to type 2 diabetes disease as well improving the performance of the older link prediction model. By combining the model which learns embedding from structure information and entity textual information.

1.1. Aims and objectives

1. To improve the existing performance of the link prediction model for drug repurposing by using knowledge graph data
2. To combine the embedding from different types of representations
3. To propose a drug list for the treatment of type 2 diabetes

1.2. The scope of work

1. Perform link prediction on dataset scraping from SemMedDB database which stores all the biomedical predication extracted from academic papers.
2. Further scraping of entity description from UMLS Metathesaurus is done to enhance the performance of the textual encoder
3. Propose a method combining different types of link prediction models from the graph structure and textual representation.
4. Evaluate link prediction performance by using several evaluating metrics to verify the performance of the model.

1.3. Research funding

I receive funding for my studies through the Graduate School at Chulalongkorn University's Scholarship for International Graduate Students program. This study is funded by the 90th Anniversary Chulalongkorn University Fund (Ratchadapiseksomphot Endowment Fund) and the 72nd Anniversary His Majesty King Bhumibol Adulyadej Scholarship.

1.4. Publication

- Mam, S., Wichadakul, D., & Vateekul, P. (2023). Drug Repurposing for Type 2 Diabetes Using Combined Textual and Structural Graph Representation Based on Transformer. IEEE Access, 11, 65711-65724. https://doi.org/10.1109/ACCESS.2023.3289863
    - IEEE Access, Institute of Electrical and Electronics Engineers (IEEE), Tier 1.
    - Impact Factor = 3.476.

CHAPTER 2

BACKGROUND

This chapter provides the background information that is related to the rest of the thesis. General knowledge of diabetes, drug repurposing, link prediction, and its evaluation methods will be discussed.

2.1. Diabetes

Diabetes is a type of long-lasting disease. Our body turns what we consume into sugar or its other name glucose. We eat to gain energy. The mechanism behind this is that our body turns food into glucose which is released into our blood. If the glucose level is high, it will trigger the pancreas to produce insulin. Insulin is very crucial as it turns glucose into energy. Hence, the abnormality of insulin happened when a person is diagnosed with diabetes either the level of insulin is too low, or our body's insulin usage is not efficient anymore. Without good insulin management or an insufficient amount of insulin to respond to the level of glucose in our blood could lead to many other malfunctioning of the body such as heart disease, vision loss, and kidney disease.

There is still no real cure for diabetes, but some activities can be taken to lessen the condition by decreasing weight, eating well, and exercising. Other measures that should be taken are to get prescribed drugs from a specialist and take the drugs regularly, educate on how to manage one's lifestyle to cope with the disease and meet the doctor as frequently as possible.

We can divide diabetes into two categories depending on the problem with our production of insulin: type 1 diabetes and type 2 diabetes. Many believe that autoimmune reaction is the main cause of type 1 diabetes which means the body makes a mistake and try to attack our own body. This type of diabetes affects our body by producing less insulin and can be diagnosed in a very early stage in humans

(kids or teenagers). However, only about 5-10% of diabetes patients have type 1 diabetes. There is currently no way to prevent this type of diabetes.

More people got type 2 diabetes which accounts for approximately 90%-95% of all diabetes patients. This type of diabetes causes our body to not use insulin as well as a normal person should. It takes years to develop but has shown more and more at early age recently. The symptoms are not very noticeable, which is why we should get our blood tested if there is a risk presented.

There are also interesting conditions of diabetes such as gestational diabetes and prediabetes. Gestational diabetes happens in a pregnant woman without diabetes but increases the baby's risk of diabetes at some point in their life. Prediabetes has also started to cause concern in adults. About 1 in 3 adults is diagnosed with prediabetes. The condition of prediabetes is very similar to those of type 2 diabetes as insulin cannot control the level of sugar in the bloodstream, but on a lower scale than those of diabetes.

2.2. Drug repurposing

Drug Repurposing or sometimes called drug repositioning, reprofiling, or re-tasking is a study on searching for already existing drugs and looking for more indications of use for other diseases. This method is effective and a lot less expensive than trying to come up with a brand-new drug which is a very long process taking lots of experiments and trials. Repurposing the original existing drug can cut down the time in a drug development process that is needed before the drug is put out such as preclinical testing, and safety. One successful example of drug repurposing is Sildenafil or more known as Viagra. Originally it was intended to treat hypertension, Pfizer repurposed sildenafil and marketed it as Viagra, giving it a market-leading 47% share of the erectile dysfunction medicine market in 2012. Global sales of Viagra in 2012 reached $2.05 billion. The first three steps of the drug repurposing strategy that need to be done before the development pipeline is taken are to find a potential chemical for a certain indication, conduct mechanistic assessments of drug effects in preclinical models, and phase II clinical studies are

used to evaluate efficacy. However, out of the three steps the first step, identifying potential drugs is the most crucial part as these candidate drugs need to be very confident to be able to continue to the next process [9].

There are multiple approaches to drug repurposing. They can be grouped into two main types: computational approaches and experimental approaches. Computational approaches are concerned with the use of any types of data which are associated with the understanding of drug usage such as gene expression, chemical structure, etc. Some popular method includes signature matching which compares a drug's unique signature with that of other types of drugs, disease, or even clinical prototype, and molecular docking, which look for the binding site between the drug and target (could be a gene or receptor) [9]. Figure 2 summarizes the different approaches to drug repurposing.

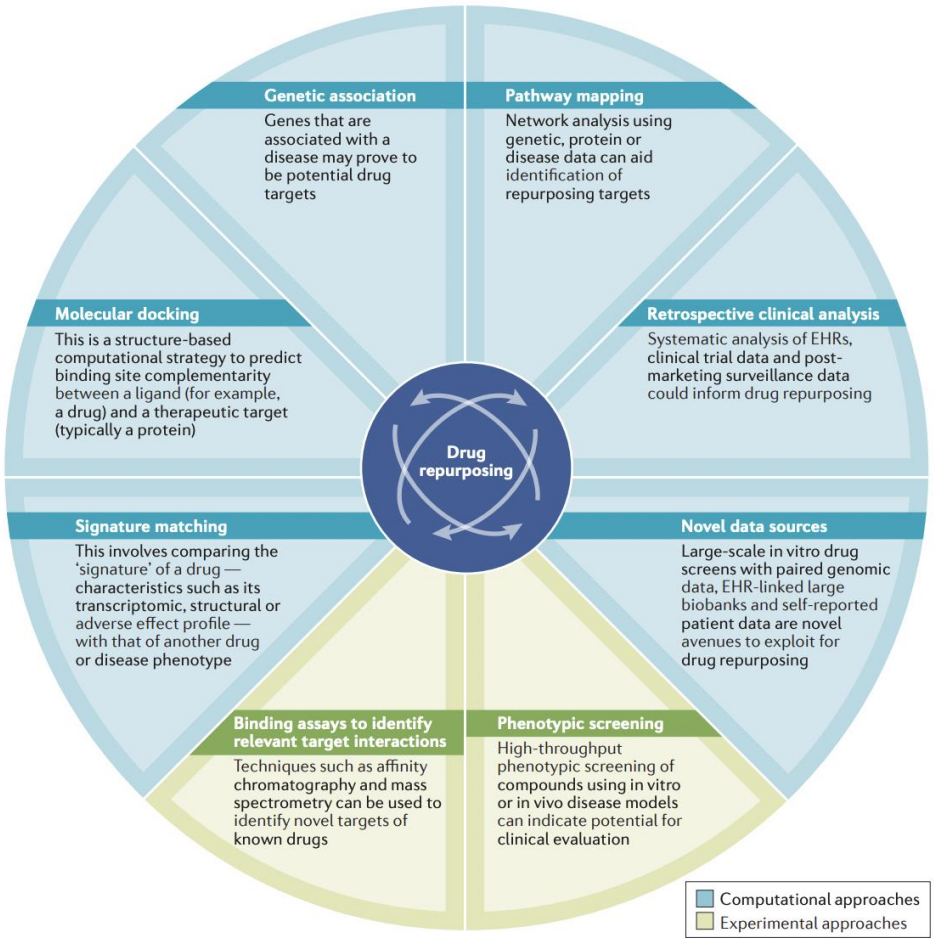


Figure 2 Different approaches to drug repurposing [9]

2.3. Link prediction

Knowledge graphs are far from complete and dirty just like any form of data. That is why knowledge graph completion comes in to fill in and fixes the incorrectness of the information. Link prediction is a task in knowledge graph completion. It works by searching for new linkages in the existing graph. Starting by learning from original data, it could infer new possible relations that may not be presented. There are many ways to achieve the task with one of the most primitive ones being latent matrix factorization as graph representation could be illustrated with a matrix. Nowadays, many studies on graphs have proposed multiple methods to produce a small-sized embedding for every node and relation in the graph, or what we can call graph embedding. The new representation encodes insightful information which could then be used for predicting new linkage. We can use the notion $\mathcal{K} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ which means that a knowledge graph is a set of entities and relations combined which consisted of three parts the head, relation, and tail. $\mathcal{E}$ is the set of all the entities while $\mathcal{R}$ is the set of relations in the knowledge graph. Link prediction is a way to complete the graph. So, the model would learn from $\mathcal{K}$ to create $s(h, r, t)$ which is the scoring function of a particular model giving a higher score to triple with high linkage possibility and keeping impossible linkage score as low as possible with $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$ mostly represent in vector in knowledge graph embedding method. By representing h, r, t as the head, relation, and tail with $\boldsymbol{e}_h, \boldsymbol{e}_t \in \mathbb{R}^{d_e}$, $\boldsymbol{e}_r \in \mathbb{R}^{d_r}$ and $\boldsymbol{e}_h, \boldsymbol{e}_r, \boldsymbol{e}_t$ represent the embedding of the three parts of the triple we can say that $s(h, r, t) = f(\boldsymbol{e}_h, \boldsymbol{e}_r, \boldsymbol{e}_t)$. So, the model of link prediction needs to be able to find the optimal solution to the f function.

2.4. Evaluation metrics

For a triple (I, k, j) in the testing set, we will calculate the ranking either by masking the head or tail part of the triple. So, to predict either the head part or the tail part of the triple we would replace each part with all the entities in the knowledge graph to create the corrupted set. We then filtered any corrupted triple that already exists in the training or the validation set. With the filtered corrupted

triples, we calculate the score for each triple and rank them from the highest to the lowest. Therefore, we can get the ranking of the ground truth for the calculation of mean rank (MR), mean reciprocal rank (MRR), and Hits@K score as represented in the formula below.

$$\mathrm{MR} = \frac{1}{2|\mathcal{K}^{test}|} \sum_{(i,k,j)\epsilon\mathcal{K}^{test}} \big(rank(i|k,j) + rank(j|i,k)\big) \tag{1}$$

$$\mathrm{MRR} = \frac{1}{2|\mathcal{K}^{test}|} \sum_{(i,k,j)\epsilon\mathcal{K}^{test}} \left(\frac{1}{rank(i|k,j)} + \frac{1}{rank(j|i,k)}\right) \tag{2}$$

$$\mathrm{Hits@K} = \frac{1}{2|\mathcal{K}^{test}|} \sum_{(i,k,j)\epsilon\mathcal{K}^{test}} \big(\mathbb{1}(rank(i|k,j) \leq K) + \mathbb{1}(rank(j|i,k) \leq K)\big) \tag{3}$$

CHAPTER 3

LITERATURE REVIEW

The focus of this chapter will be on the related model for the thesis including the broad concept of deep learning and its variations technique which are used to apply the task of link prediction. After that, we will discuss several different approaches such as the traditional method, graph convolutional network, graph attention network, and transformer-based graph embedding.

3.1. Deep learning

Deep learning is a part of machine learning algorithms that focus more on trying to imitate how neurons work in the human brain. It consists of nodes connecting with multi nodes on multilayers which is where the word deep comes from as the number of layers is dependent on the specific design of the different models and some models can have a very steep network. Deep learning can learn to extract useful features from the data without the need for humans. Additionally, the success of deep learning also owes to the increase in the amount of data that we could acquire in this age. These factors have gathered more interest in deep learning models and have the effect of improving the performance of many underperformed tasks immensely such as speech recognition, image classification, and image segmentation. As the study of deep learning became widespread many types of models are proposed for solving different domain-specific problems. Some examples include Convolutional Neural Network and the Transformer model.

3.1.1. Transformer

Transformer, from the paper “Attention Is All You Need” from Google with its highly improved architecture, has transformed the deep learning field forever. The model consisted of two parts: an encoder and a decoder. As with most tasks in NLP,

the input of the model is a sequence of text, and the output produces a sequence of output. The encoder encodes the input while the decoder receives the representation from the encoder and transforms it into the output. The backbone of the transformer model is mainly based on the attention mechanism which is defined in the equation below. With self-attention, the model would be able to attend to some parts of the input more than others as not all segments of the inputs are equally relevant.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

The whole transformer architecture consists of self-attention as well as pointwise put on top of each other multiple times with similar architecture for both the encoder and encoder model [4]. Although, the architecture in Figure 3 contains both the encoder and decoder we can also have a transformer composed only of the encoder or decoder by itself for example the Bidirectional Encoder Representations from Transformers (BERT) consists only of encoders while the popular text generation model GPT may include only decoders in its architecture. Transformer has become widely popular and found new usage in other fields besides NLP as well such as in computer vision with the Vision Transformer model [10]. Because of its flexibility and high performance, the transformer model has been used in a variety of domains and became the state-of-the-art model for many problems.

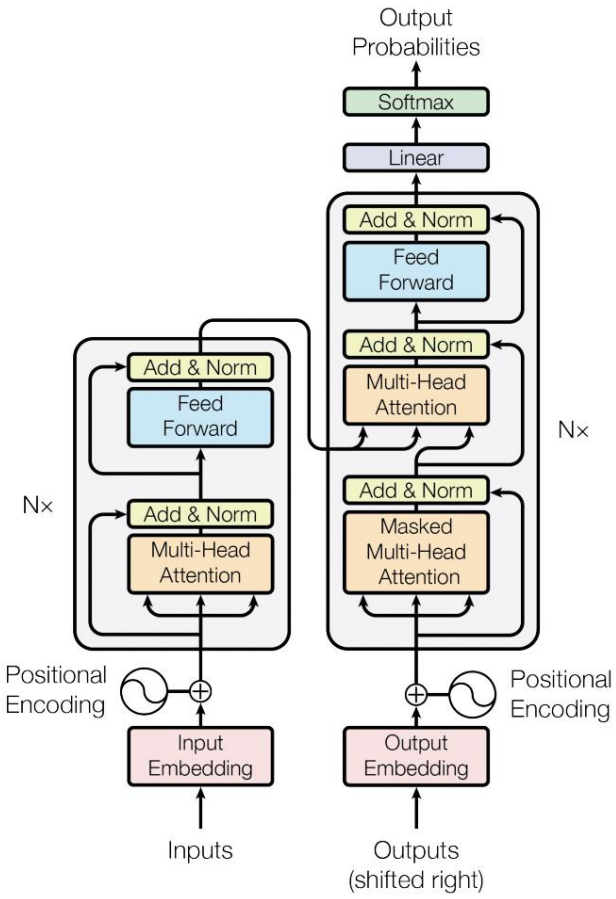


Figure 3 Transformer architecture [4]

3.1.2. Bidirectional encoder representations from transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) is a type of transformer with only the encoder as the component. This type of transformer is very popular for generating embedding for downstream tasks. BERT is provided as a pre-trained model on a large corpus with each model that could be trained on the varying type of domains that we can choose from to accommodate the purpose and objective of our job. The pre-trained process provides the base for creating rich embedding which is trained with unsupervised tasks by using a masked language model, where some words are left out for the model to predict or the next sentence prediction which are important for finetuning of tasks like question answering and natural language inference which it is crucial to understand the nature and the relation of multiple sentences [5]. As shown the Figure 4, the BERT model begins

with the pre-training phase which the model is trained to predict the masked word of the masked language model. After completing the first step we can use the pre-trained model on our downstream tasks such as Named-Entity Recognition. In BERT some special tokens are also included. [CLS] token is used at the beginning of the sequence as a global representation while [SEP] is a special token for separating different sections of input e.g., in question answering task we could use [SEP] as a separator between the question-and-answer part of the output.

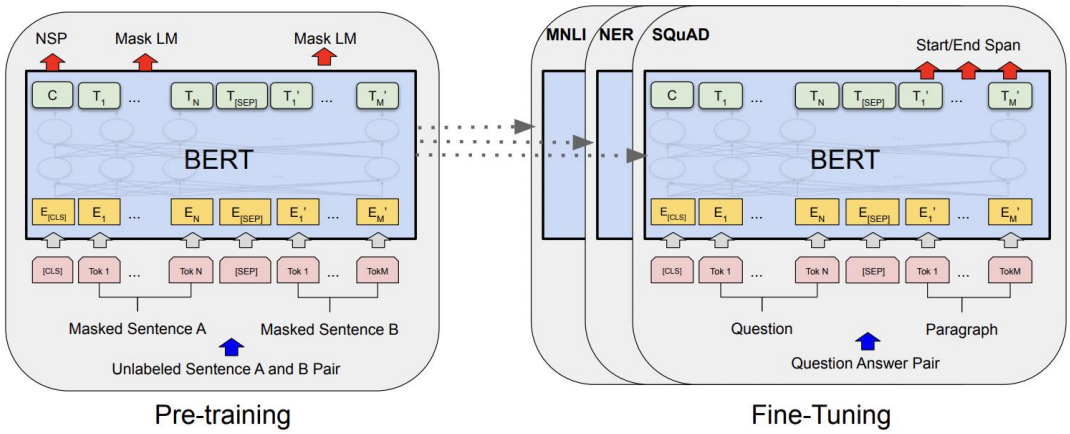


Figure 4 Bidirectional Encoder Representations from Transformers [5]

3.2. Knowledge graph representation

To get the most benefit from knowledge graphs, they need to be represented in a way that computers can process and learn from them. Most knowledge graph representation involves embedding entities and relations into small-sized vectors. Many methods are employed to achieve this from the simplest to the most sophisticated model.

3.2.1. Traditional graph embedding

In most traditional graph embedding, the scoring function is used to learn the embedding of both the entities and the relation. As the entities with head and tail parts are connected by a relation, embedding can also be translated from how all the entities are attached in the graph. For example, in the simplest graph embedding

model TransE, it is hypothesized that from a triple (h, r, t) the sum of the head (h) and relation (r) embedding should be as close to the embedding of the tail embedding as possible. Hence, in link prediction, any (h, r, t) triples with the highest scoring function will have the highest probability of being a real triple. The scoring function of various traditional graph embedding is listed in Table 1.

Table 1 Scoring function of different graph embedding models [11]

Model	Scoring Function $\boldsymbol{f(h,r,t)}$
TransE [3]	$-\|h+r-t\|_{\frac{1}{2}}$
TransR [12]	$-\|M_r h + r - M_r t\|_2^2$
DistMult [13]	$h^T diag(r) t$
ComplEx [14]	$Real(h^T diag(r) \bar{t})$
RESCAL [15]	$h^T M_r t$
RotatE [16]	$-\|h \circ r - t\|_{\frac{1}{2}}$

3.2.2. Graph convolutional networks and graph attention networks

There is also an attempt in implementing a convolutional neural network (CNN) on graph embedding which is introduced in the paper [17] Graph Convolutional Network or GCN. While CNN is designed for image data GCN is designed so that it would work with graph data. In CNN, the kernel of fixed is applied to the image by performing a calculation on each window of the segment of the image one by one until it covers all the pixels in the image. In other words, in each window, the pixels that are next to each other are calculated with the kernel and are averaged to get the numeral representation of that window. Similarly, in graph data, each node is accompanied by several adjacent nodes which are connected by relations. To get the embedding of each node the embedding for the node itself is

combined with the embedding of the neighbor node and averaged to create the final embedding. Figure 5 illustrates the similarity between GCN and CNN.

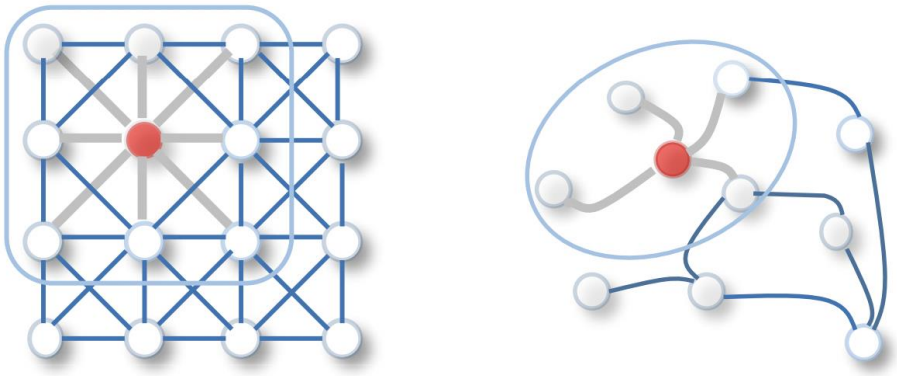


Figure 5 CNN and GCN comparison [18]

Another graph embedding technique called, Graph Attention Networks (GAT) which is an improvement from GCN, assigned an attention score to the neighbor node so that the most important neighbor node would have the highest effect in the calculation of the node embedding. Both techniques mentioned above have become the standard in graph embedding for many tasks. However, as you may have noticed these models do not consider relation embedding while calculating the node embedding.

Although graphs can consist of only graphs with nodes attaching most knowledge graphs are real entities connected by the relation of multiple types. So, if any two nodes are connected by one type of relation they may or may not be also connected by another relation type as well which is why the relation type embedding is very crucial in the representation of a knowledge graph. In [19], the authors integrate the power of GAT with the embedding of relation type by concatenating the head, tail, and relation together into one long vector and assigning different attention weights to all the triples in the knowledge graph. Same with GCN, neighboring nodes still play an important role in embedding the knowledge graph. The model considers the context of a node with more than just the immediate neighbor node but also the next hop as shown in Figure 6 the embedding of the

node U.S. is calculated from 2 hop neighbors which could provide even more rich embedding as more information is learned from the farther node.

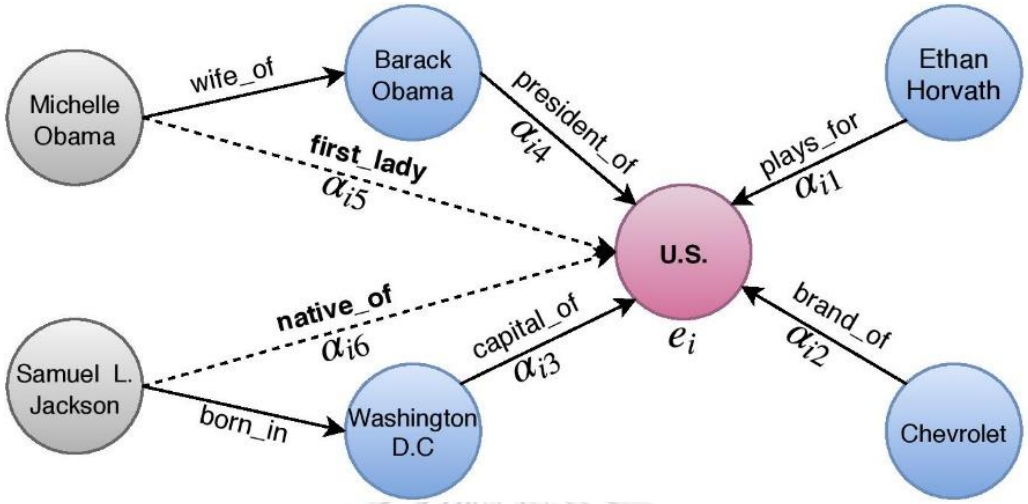


Figure 6 Multi-hop neighbor in KBAT

3.2.3. Graph textual embedding approach

Most knowledge graph entities and relations also have their correspondent's real-world name which could be used as the textual representation of natural language text. For example, the entity "Joe Biden" has the relation "Is president of" with the entity "USA". As mentioned in (3.1.2), with a pre-trained BERT model these texts could also provide information from their textual meaning in human language. In [6], the authors proposed a model called KG-BERT, which makes use of a language model on knowledge graph tasks. Entities and relations are represented in text and put together in a sequence, as shown in Figure 7 The model helps improve the previous by increasing the robustness of the model which was suffering from unseen entities. However, KG-BERT uses a single transformer model and suffers from high overload since in the inference stage it must calculate all the possible combinations of entities and relations which results in slow running time.

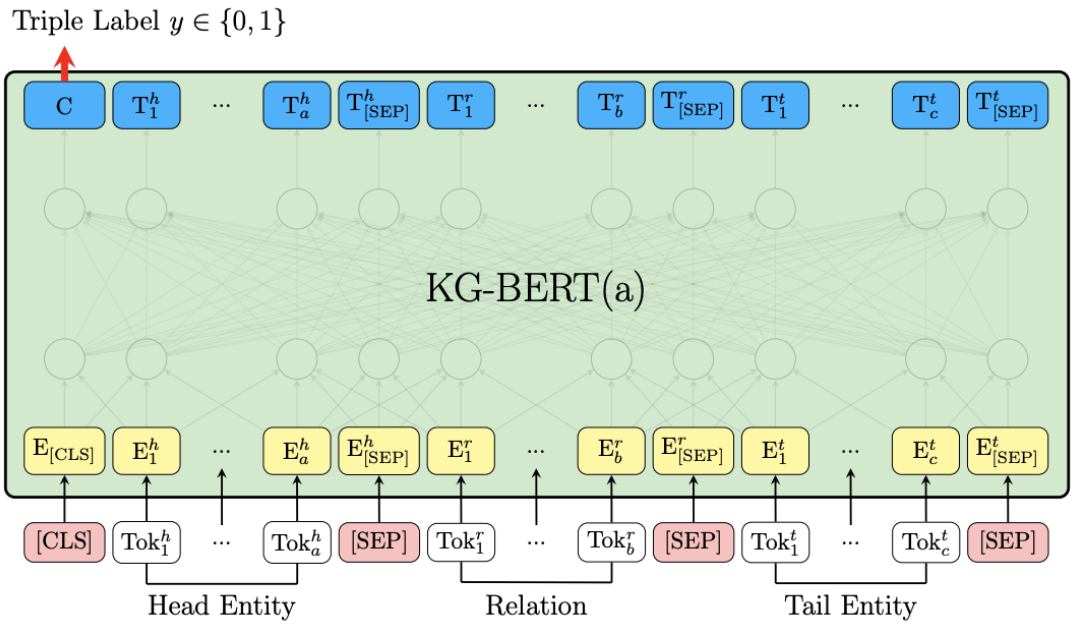


Figure 7 Architecture of KG-BERT [6]

The model named "Structure-Augmented Text Representation" (StAR) was proposed to fix the problem faced by KG-BERT. The architecture of StAR is illustrated in Figure 8. In StAR, the triple is divided into two separate parts. The first part is the concatenation of the head entity and relation text. The second part consists only of the entity text of the tail section of the triple. By doing this, it reduces the number of combinations as seen in KG-BERT which must feed all three parts of the triple in one go. The model parts are twisted by adding a pair of BERT models instead of only a single model. StAR uses a Siamese network with shared weight. Hence, the first model is fed with the head and relation concatenation part, and the second part is fed with the tail entity text. After that, the output of both parts of the triple is used to compute two loss scores. The first loss score is the classification objective, which follows a similar approach to other NLP tasks and implements multi-layer perceptron and utilizes binary cross entropy loss to produce the label if the triple combination is possible or not (1 for real triple and 0 for negative triple). The second training objective called the contrastive objective uses scoring like traditional graph embedding methods by finding the distance between the head and relation part with the tail part so that the negative triple would have a bigger distance compared

to the positive triple. StAR proves to be very efficient and performs much faster than KG-BERT and the model performance also improves.

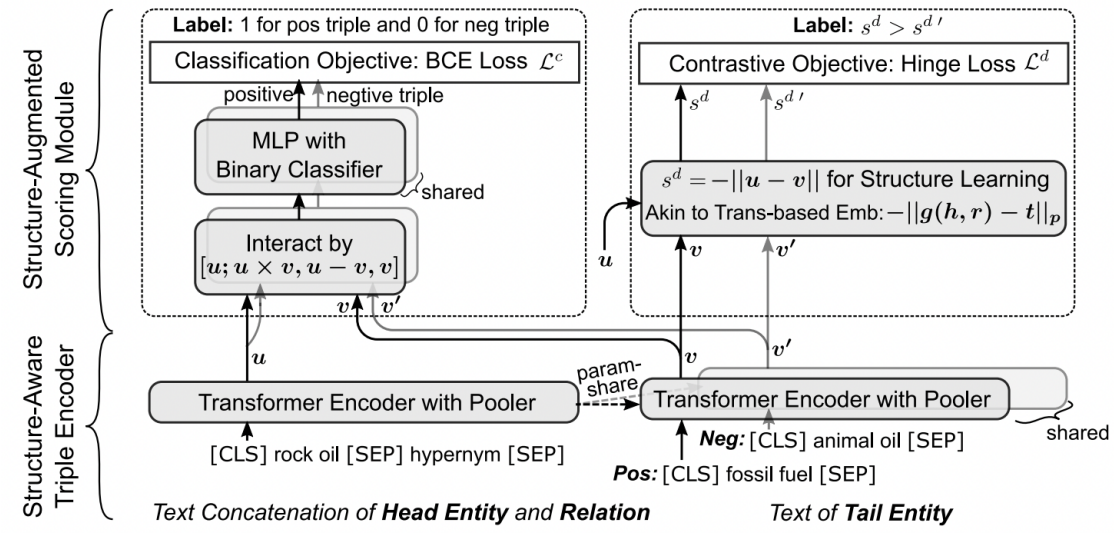


Figure 8 Architecture of StAR model [7]

3.2.4. Graph structural embedding approach

HittER is another transformer-based graph embedding model. The difference between HittER and KG-BERT and StAR is that the transformer model used in HittER is not a language model. As most language models use the token of textual data as the input Hitter treats every single entity or relation as one token to train the transformer model. As seen in Figure 9, HittER is a hierarchical model which means there are two blocks of transformers placed on top of each other. The first block creates head relation-specific embedding by incorporating the head and relation of the triple as the input while the second block combines the embedding of the main triple and its neighbor. The possibility score of the triple is computed by the dot-product of the output of the second block with the embedding of the tail entity. Therefore, HittER not only learns the representation at the triple level but also includes information from the context which in this case is the neighbor nodes of the head embedding.

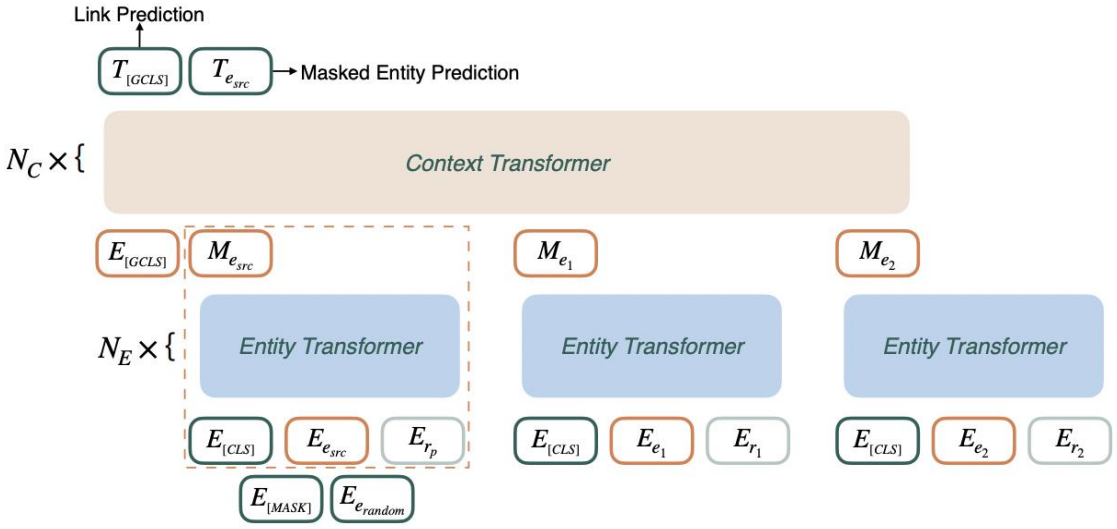


Figure 9 HittER architecture [8]

CHAPTER 4

CONCEPT AND RESEARCH METHODOLOGY

In this chapter, the model concept proposed for improving the link prediction task for drug repurposing is discussed. As discussed in the last chapter, there are two types of transformer-based models one focuses on nodes and relations textual representation while the other one is to learn the contextual embedding from surrounding nodes. For the model in this thesis, we would like to propose a way to combine both representations from the already existing model to further increase the performance of the link prediction model.

4.1. Data acquisition

Most of the data acquisition process is replicated from the paper [3] which is a paper on drug repurposing for Covid-19. The input data of this thesis is a knowledge graph extracted from the research literature database. The overview of the whole data is illustrated in Figure 10.

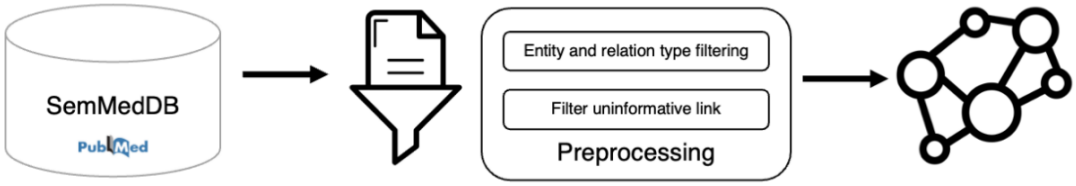


Figure 10 Data preparation process

4.1.1. SemMedDB

All the triples data used in this thesis is provided by SemMedDB [20]. SemMedDB is a database that contains all the predicates (relations) in the biomedical field extracted from PubMed citations (approximately 29.1 million citations) by using NLP tools called SemRep (rule-based relation extraction). A

summary of how SemRep works is it maps all the entities appearing in a sentence to UMLS Metathesaurus, another database that keeps track of biomedical field vocabulary. After that, it will look for how those entities interact in the sentence and normalize the relation to the standardized relation name of Semantic Network (a standard controlling the schema of one entity interacting with another) since there may be multiple ways to say the same word. Further preprocessing is needed to clean useless triples and limit the number. The latest version of the raw dataset that was extracted from SemMedDB contains a total of 116,603,760 predicates.

4.1.2. Preprocessing

We began by removing the first class of unnecessary entity types in SemMedDB called generic biomedical concepts. The list of generic concepts is pre-defined in one of the tables in the database of SemMedDB and includes entity types like Pharmaceutical Preparation. Then, we consulted with the domain experts to select the most useful relation type for our drug repurposing task. We were provided with a list of 20 relation types. Additionally, any entities with the entity types of Activities & Behaviors, Concepts & Ideas, Objects, Occupations, Organizations, and Phenomena were also discarded from the dataset. After the filtering of this step, 54,735,504 predicates are left for further processing.

The next filtering involves removing uninformative links which use statistical value as the threshold. The value is calculated by using network degree centrality to filter out high-degree concepts and log-likelihood ratio for any triples which do not contain useful information. To calculate the degree of an entity, we use an adjacency matrix which is one way to present a knowledge graph. The rows and columns of the matrix represented the entities in the knowledge so that if there is a connection between any two entities the cell A_{ji} would equal 1 (i and j correspond to the first and second entity). Then, the in and out degrees of concept i can be calculated by using the following formula.

$$k_i^{in} = \sum_{j=1}^{n} A_{ji} \tag{5}$$

$$k_i^{out} = \sum_{j=1}^{n} A_{ij} \tag{6}$$

As for uninformative links calculation, we used G^2 score which is the indicator of how strongly the three terms (head, relation, tail) are linked together. If the observed and expected frequencies differ significantly, the triple is less likely to happen by random and has a high G^2 score. The formula which is used to calculate the score is as follows.

$$G^2 = 2 \times \sum_{i,j,k} n_{ijk} \times \log\left(\frac{n_{ijk}}{m_{ijk}}\right) \tag{7}$$

$$m_{ijk} = \frac{\sum_i n_{jk} \times \sum_j n_{ik} \times \sum_k n_{ij}}{T^2} \tag{8}$$

n_{ijk} is the frequency of the term i, j, k appearing together, and T is calculated by $T = \sum n_{ijk}$

All three measures mentioned above (G^2, k_i^{in}, k_i^{out}) were normalized to the range of 0 to 1 and summed up together to produce the final score. The score indicates that the more specific the relation the lower the score. A group of an entity related to diabetes was all kept in the dataset without any filtering. (Refer to the list in APPENDIX A). Also, with the limitation of GPU and model size, we only kept only an adequate number of triples for training which result in the final number of triples in the dataset being 81,008 triples after removing duplication and more entity type filtering.

4.1.3. Time-slicing data split

To test the ability of our model performance we divided the dataset into training, validation, and testing sets. We split the dataset in the same way as a time

series dataset. We use the data from before 2021 as the material to train our model while testing it on the future triple. This is an imitation of a real-case scenario to see if the knowledge that is available to us at the current time could be used to predict the drug which can be a possible treatment for the future. Duplicate predicate from different timeframe is also deleted in this step. As a result, the final number of examples in each set is 64,830, 6,215, and 9,963 triples for the training, validation, and testing sets respectively.

4.1.4. Entity description

As shown in Table 2, the dataset consists of 26,598 entities with 18 relation types. The top 3 relation types with the most triples are INTERACTS_WITH, COEXISTS_WITH, and AFFECTS. TREATS, which is the main type for identifying treatment in drug repurposing tasks, accounts for 6.77% of the data. To enhance the quality of the textual encoder module, we incorporated additional entity descriptions for link prediction. The descriptions were scraped from UMLS Metathesaurus through rest API. We were able to scrape descriptions for 17,421 out of 26,598 entities. As for entities without descriptions, only the entity names will be used.

Table 2 Relation Name

Relation Name	Inverse Relation Name	Count (%)
INTERACTS_WITH	INTERACTS_WITH	25.71
COEXISTS_WITH	COEXISTS_WITH	14.60
AFFECTS	IS_AFFECTED_BY	9.59
ASSOCIATED_WITH	ASSOCIATED_WITH	8.41
INHIBITS	IS_INHIBITED_BY	7.87
CAUSES	IS_CAUSED_BY	7.43
TREATS	IS_TREATED_BY	6.77
STIMULATES	IS_STIMULATED_BY	6.67

PREDISPOSES	IS_PREDISPOSED_BY	3.89
DISRUPTS	IS_DISRUPTED_BY	2.36
PREVENTS	IS_PREVENTED_BY	1.80
AUGMENTS	IS_AUGMENTED_BY	1.72
PRODUCES	IS_PRODUCED_BY	1.62
COMPLICATES	IS_COMPLICATED_BY	0.50
USES	IS_USED_BY	0.49
PRECEDES	IS_PRECEDED_BY	0.33
MANIFESTATION_OF	IS_MANIFESTED_BY	0.14
CONVERTS_TO	IS_CONVERTED_FROM	0.09

4.2. The combined model method

We used two models of graph embedding as the base for the combination. The text or the description if implemented in pair with pre-trained language can immensely provide deep meaningful semantic information about the entities and relations. However, we cannot ignore the neighbor node size of information which is another useful source. Hence, in this section, we will describe the two base models and how we will combine them to improve performance. Table 3 compares the definition and the description of the two types of graph embeddings.

Table 3 Comparison of textual and structural graph representations along with aspects such as node representation, input, and training method.

Topic	Textual representation	Structural representation
Node representation	Each node and relation can be represented with meaningful words or phrases.	Each node is connected to its neighbors via different types of relations.
Input	The combination of head, relation, and tail in its textual form is used to learn from.	Utilizing information from nearby nodes, node embedding can be enriched.

Topic	Textual representation	Structural representation
Training method	Can be seen as a possible sentence classification where each word/token provides the semantic information and collocation of surrounding words.	With information about its neighbors, the model can learn clues to predict similar relations with relatively similar structured neighbors.

4.2.1. Structure-augmented text representation (StAR)

The triple is split into two separate components in StAR [7]. The head entity and relation text are concatenated to form the first component. The entity text for the triple tail segment is included in the second component. The Siamese network used by StAR consists of two BERT models with shared weights. As a result, the first model receives the head and relation concatenation, while the second receives the tail entity, both in text form. First encoder module inputs are as follows: $x^{[CLS]}, x^{(h)}, x^{[SEP]}, x^{(r)}, x^{[SEP]}$second encoder module inputs are $x^{[CLS]}, x^{(t)}, x^{[SEP]}$. To make a more complex embedding, the head and tail input can also include their description, adding extra definitions for each entity input. There are no negative examples in the knowledge graph. However, it is essential to sample negative cases to train the model. By introducing some random entities into one of the triple's entity parts, negative sampling is accomplished. If the graph's positive triple is denoted by $tp = (h, r, t)$, the negative triple is denoted by $\{(h', r, t) | h' \in \mathcal{E} \wedge (h', r, t) \notin \mathcal{K}\}$ or $tp' \in \{(h, r, t') | t' \in \mathcal{E} \wedge (h, r, t') \notin \mathcal{K}\}$, respectively, depending on whether the corrupted triple is in the head or tail of the original K.

After that, two loss scores are computed using the output from both sections of the triple. The classification objective is utilized as the first loss. The classification objective uses a multi-layer perceptron with a binary cross entropy loss to determine if the classification of the combination is correct or not: 1 for a positive triple and 0 for a negative triple. This approach is like that of other NLP tasks. The following formula can be computed:

$$\mathcal{L}^c = -\frac{1}{|\mathcal{D}|}\sum_{tp\in\mathcal{D}}\frac{1}{1+|\mathcal{N}(tp)|}\left(\log s^c + \sum_{tp'\in\mathcal{N}(tp)}(1-\log s^{c'})\right) \quad (9)$$

where $\mathcal{N}(tp)$ is the number of negative triples for every positive triple and $\mathcal{D}$ is the total number of positive triples. The scores or probabilities of the positive and negative triples, respectively, are denoted by s^c and $s^{c'}$.

The contrastive objective is the name of the second training objective. By measuring the distance between (1) the head and relation component and (2) the tail part, the contrastive objective employs scoring like conventional graph embedding methods to ensure that a negative triple has a greater distance than a positive triple. For the positive triple tp and the negative triple tp', the distance-derived score is denoted with s^d and $s^{d'}$, respectively. The following is an example of a margin-based hinge loss:

$$\mathcal{L}^d = \frac{1}{|\mathcal{D}|}\sum_{tp\in\mathcal{D}}\frac{1}{|\mathcal{N}(tp)|}\sum_{tp'\in\mathcal{N}(tp)} max\left(0, \lambda - s^d + s^{d'}\right) \quad (10)$$

The final learning objective's propagation and training are done at the same time. The following is the definition of the two-loss equations' sum:

$$\mathcal{L}^d = \mathcal{L}^c + \gamma\mathcal{L}^d \quad (11)$$

where γ represents the weight.

Figure 11 depicts an overview of the entire textual model. The transformer encoder model's output provides the embedding vector to the loss module, which may be thought of as a decoder. The score for the triple in link prediction can be produced by either one of the two losses. Our initial study shows that both losses produce comparable results. The architecture's fine-tuned encoder module can be utilized to produce the embedding for textual representation in the proposed model once the pre-trained model has been trained using the link prediction task.

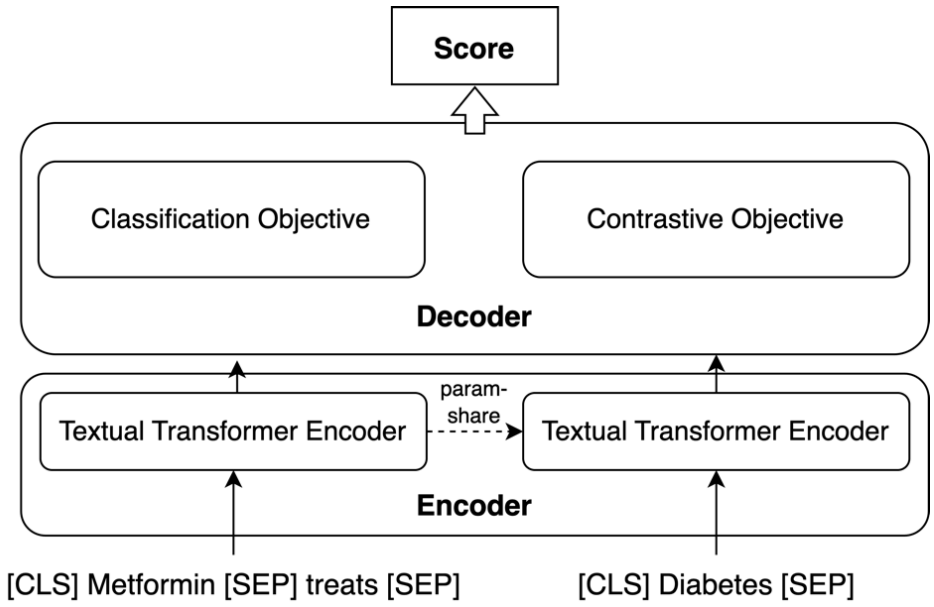


Figure 11 Architecture of textual model (StAR) [7]

4.2.2. Hierarchical transformers for knowledge graph embeddings (HittER)

Another transformer-based graph embedding approach is HittER [8]. The transformer model used in HittER differs from those used in KG-BERT and StAR in that it is not a language model. Textual data is the primary input for most language models. In contrast, HittER trains the transformer model by treating each entity or relation as a single token. Since HittER uses a hierarchical framework, two blocks of transformers are stacked one on top of the other in the encoder. The head and relation of the triple are used as inputs in the first block to generate the head relation-specific embedding. The embedding of the primary triple and its neighbors are combined in the second block. The dot-product of the second block's output and the tail entity's embedding is used to calculate the triple possibility score. Since the neighbor nodes of the head embedding are the context in this instance, HittER not only learns representation at the triple level but also incorporates context-related data.

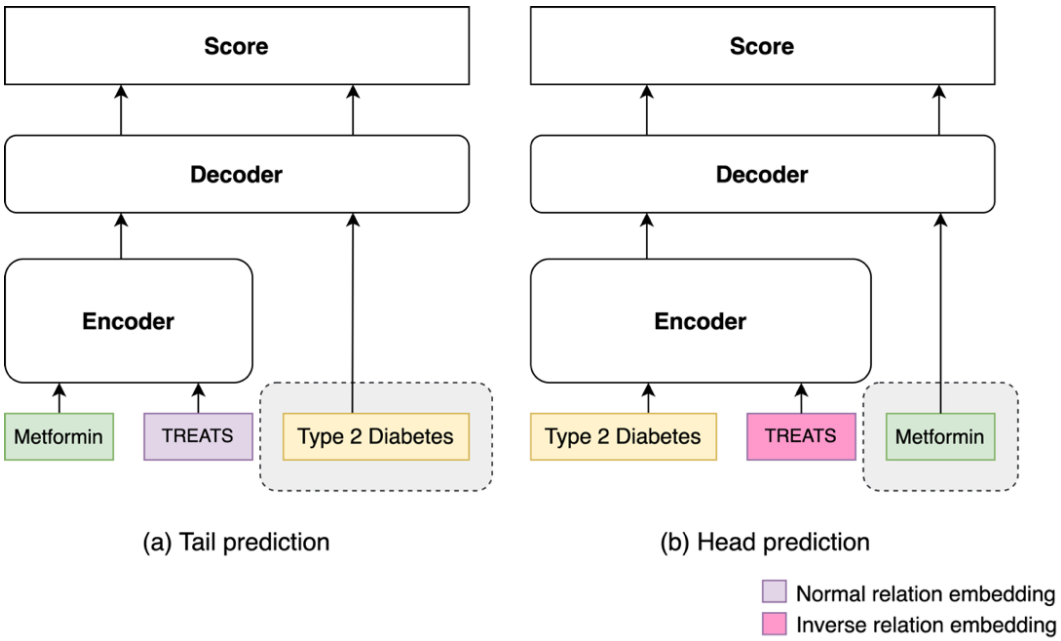


Figure 12 Comparison of (a) tail prediction and (b) head prediction in the HittER mechanism. For data input, the triple (Metformin, TREATS, and Type 2 Diabetes) is used as an illustration. The head and relation are inputted into the encoder for the tail prediction task, whereas the tail and relation are input into the encoder for the head prediction task. Utilizing different values for relation embedding, in which various colors are employed to represent different relation inputs.

HittER uses two different scoring types for head and tail prediction, as shown in Figure 12. Both the head and the relation must be provided into the encoder for it to forecast the tail. Both the tail and the relation must be provided into the encoder for it to forecast the head. The effectiveness of the model can be increased by dividing the two scoring functions for prediction [21, 22].

4.2.3. Proposed method

The vector representation of the textual side is created using StAR. We can benefit from the pre-trained embedding of biology text by using BioBERT as the pre-trained model. StAR models are trained in two different ways. We will fit the original data in its original order into the model in the first one, which is called Normal StAR. While the second, called Inverse StAR, will be input with the triple's reversed form.

As an example, a triple “A treats B” will be changed to “B is treated by A”. We obtain the embedding matrices for the head part (entity and relation) and the tail part, which will be further implemented in the following phase, after training both models. Both models are trained using the BioBERT pre-trained transformer model, which was created using a sizable biomedical corpus, including PubMed Abstracts (4.5 billion words) and PMC full-text articles (13.5 billion words) [23].

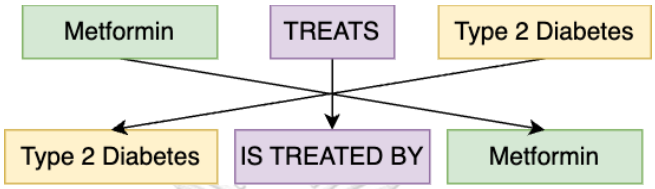


Figure 13 StAR inverse dataset reversion

We employed another transformer-based graph embedding called HittER to take the structure information into account. HittER predicts the tail part (A treats ___) and the head part (B is treated by ___) via different scoring functions. Therefore, StAR requires two separate embeddings. For combining with predicting the tail and head score functions of HittER, respectively, the normal StAR and inverse StAR will be used. The inversed process is shown in Figure 13. For example, the relation between “Metformin treats type 2 diabetes” and “Type 2 diabetes treats Metformin” must be changed for the language model to understand the new, altered triple. For instance, the passive form of the original relation IS_TREATED_ BY is the inverse counterpart of the relation TREATS. All remaining relations follow a similar process, except a few, like COEXISTS_WITH, where reading in any direction does not affect the meaning. Both the relation's original text and its inversed form are listed in Table 2. After training, the two StAR models are applied to the source component (entity and relation) as textual embedding generators or textual encoders.

HittER serves as the foundation of our model combination. As a result, inputs are initially initialized as embedding vectors made up of three tokens: the relation (E_{r_p}), the source entity ($E_{e_{src}}$), and the [CLS] special BERT token ($E_{[CLS]}$). The exact same procedure is applied to neighbor nodes for source entities and their relationships. It is possible to manually set the number of neighbor nodes, which is

denoted by (N_e). The token [CLS], which is used to represent both the source entity and relation in pair, is then produced by the first layer of the BERT entity transformer. The context transformer is a second transformer block that takes as input the output [CLS] tokens of the source and neighbor entities. The source triple and its neighbors' final embedding representation are created by this transformer block. The [GCLS] special token is added before the [CLS] token outputs from the previous transformer block. The global embedding is learned by [GCLS] from the remaining tokens.

The [GCLS] output of the structural encoder will be instantly concatenated with the embedding of the source and relation parts of the textual representation. The training process begins with the tail prediction, where textual representation for concatenation is created using the normal pre-trained textual transformer encoder. An inverse pre-trained textual transformer encoder is then used in the head prediction stage. The embedding will then use the input text “Metformin treats” for the triple (Metformin, TREATS, and Type 2 Diabetes), using the normal textual transformer encoder. In contrast, the text will be entered into the inversed embedding generator as “Type 2 diabetes is treated by”. A more thorough explanation of the two different scoring functions is provided in Section 4.2.2. It should be noted that the textual encoders only provide the textual embedding, and their weights are not learned in the combined model training process.

After concatenation, the vector goes through fully-connected layers. As such, the model can capture the information from both textual and structural representation. After that, the link prediction score is calculated using the new representation-combined embedding. The true triplet's plausibility score is calculated as the dot-product of $T_{[GCLS]}$ and the target entity's token embedding via the decoder. The plausibility scores for all other candidate entities are calculated in the same manner. Then, they are normalized using the softmax function. Finally, the cross-entropy loss is obtained using normalized distribution:

$$\mathcal{L}_{LP} = -\log\left(e_{tgt}|T_{[GCLS]}\right) \tag{12}$$

Masked entity prediction (MEP) was also used to help lessen the overwhelming noise of the surrounding nodes. The already-high-quality input may at times disregard the newly added context vector. Forcing the model to learn new knowledge could require more work. Another unique token [MASK] was added to help solve this issue. The position of the source entity is where this token is randomly substituted. As a result, noise is added to the input, forcing the model to take into account contextual information when learning. Masked entity prediction uses the output of the second block of the original source embedding, $T_{e_{src}}$. By doing this, the additional context vector won't have a significant impact on the quality of the source embedding, making it impossible for $T_{e_{src}}$ to predict its original self. The accumulation of $\mathcal{L}_{LP}$ (link prediction loss) and $\mathcal{L}_{MEP}$ (masked entity prediction loss) results in the modification of the loss as follows:

$$\mathcal{L}_{MEP} = -\log\left(e_{src}|T_{e_{src}}\right) \tag{13}$$

$$\mathcal{L} = \mathcal{L}_{LP} + \mathcal{L}_{MEP} \tag{14}$$

The proposed model architecture is visualized in summary in Figure 14.

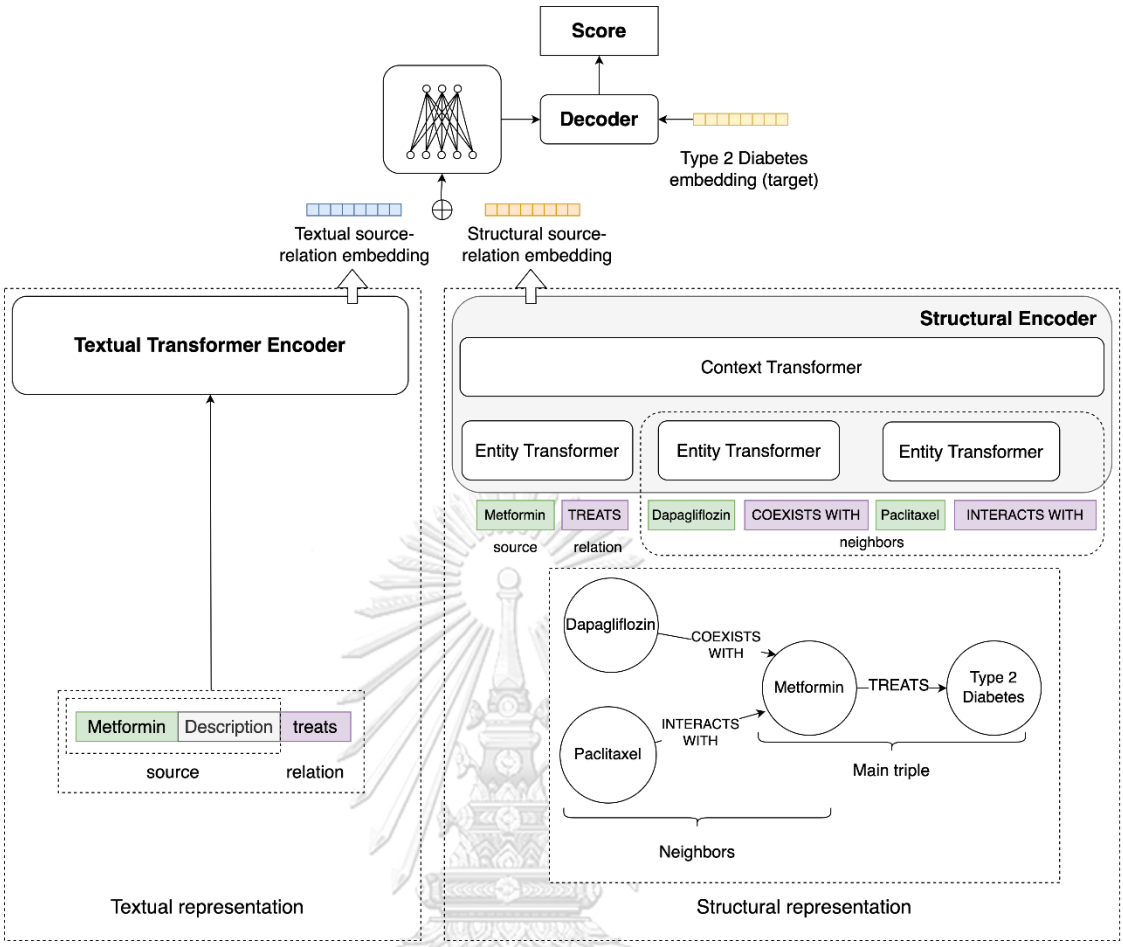


Figure 14 Proposed architecture for the link prediction model that combines textual and structural representation.

4.3. Evaluation

We will utilize mean rank (MR), mean reciprocal rank (MRR), and Hits@K to evaluate the whole test set, as is common practice in link prediction tasks. The top K values of the Hits@K score will be 1, 3, 10, and 100, even though the real-world drug list for testing can reach 500 substances. In contrast to MRR, which is more sensitive to high-ranking scores, MR is more sensitive to low-ranking values (bottom). Due to our greater interest in placing the right medicine at the top of the list, MRR offers better insight into the model performance. However, MR is a value that is easier to comprehend and understand. Additionally, it's crucial to note that, in contrast to the other metrics, MR measures performance as a function of value, where a lower value

indicates greater performance. All metrics can only reach a maximum value of 1. The quantitative evaluation process makes use of these metrics. We forecast the top 10 high-potential drugs to further evaluate the model's qualitative performance.

4.4. Drug List

Following model training, the drug list can be easily acquired by utilizing the model prediction method. The dataset contains a list of all the Pharmacological Substances that could be used as potential medications. Then we can add the drug entities to the triple "___ TREATS Type 2 Diabetes" in the head portion. The plausibility score for each medicine as well as the pairs for TREATS and Type 2 Diabetes was then determined using the inference model. By doing this, we may order the medicine from most to least probable. In our graph, there are 25,598 entities. Only the top-ranked drugs, however, will be subjected to additional testing. Given that not all entities are connected to drugs, it is advisable to get rid of them from the drug list. This list serves merely as a preliminary shortlist. To undertake a more thorough study procedure, a drug domain specialist is required. However, by using the model we can save time by removing the need to concentrate on drugs with low potential.

4.5. Experimental setup and model parameters

BioBERT version 1.1, a pre-trained model from huggingface, which keeps a variety of pre-trained models and datasets, was used for the textual transformer model [24]. For contrastive objectives, Euclidean distance was used with a learning rate of 10^{-5} for the textual module with a batch size of 40, which was the maximum number that our GPU could handle. The learning rate for the suggested model was set to 10^{-1}, and the activation function was chosen to be the Gaussian error linear unit (GEULU). A batch size of 128 was used for the model's training. The combined method used an embedding size of 320. Two dense layers were used to integrate the two representations.

CHAPTER 5

EXPERIMENTS AND RESULTS

This preliminary result from the combined model from Chapter 4 is compared with other baseline models. After learning about the training set, the performance of the model is evaluated by using the metrics discussed in 4.3.

5.1. Result

5.1.1. Overall result

A comparison of the performance of our suggested model and the baseline is shown in Table 4. We used the StAR model, a text-only graph embedding encoder model, and the HittER model, a structural information graph embedding encoder model, as our baseline models for this study. The best MRR was demonstrated by our suggested model when compared to the baseline models. Metrics for MRR showed a rise from 23.84 to 27.19. HittER confirmed its significance as the foundation of our model by outperforming all other baseline models. The lowest MR was shown by StAR inverse, giving it the top overall rating. Since inverse rank value gives more weight to models that correctly rank triples higher, MRR is advised as the primary metric. In all Hits@K measures, our model performed better than all baseline models, especially for the Hits@3 metrics. These results thus emphasize the beneficial effects of adding textual representations to triples that are already highly scored by the HittER model. As a result, using our approach, the initial scoring by HittER was elevated even more. On the other hand, the performance improvement diminished as K's value grew. These results provide empirical support for the model's two-representation encoder design. The model is found to perform better when many representations are added to it rather than depending just on one representation. As related nodes frequently display comparable patterns with one another, structural representation gives information on node interactions with its

neighbors in the graph dataset. On the other hand, the entity descriptions provide additional information about how things act in the real world, and the textual encoder teaches the model its semantic meaning. That such an improvement may be observed when applied to the same architecture is not unexpected.

Table 4 Performance of the test set using the prediction model. The performance score with the highest rating is bolded.

Model	MR (↓)	MRR(↑)	Hits@1(↑)	Hits@3(↑)	Hits@10(↑)	Hits@100(↑)
StAR	**4,203.58**	13.32	6.73	15.63	27.01	45.46
StAR inv	4,650.22	16.43	10.10	19.04	28.58	45.91
HittER	4,479.08	23.84	15.59	27.48	41.73	49.87
Ours	4,502.62	**27.19**	**18.18**	**33.23**	**43.17**	**50.11**

5.1.2. Treatment-task result

The main goal of our study is medication repurposing. We only considered the TREATS relation while evaluating the model's performance. The efficacy of the model was subsequently examined to find triples that were connected to the treatment. Table 5 shows that our model beat the typical HittER model across all metric scores, which is noteworthy. The effectiveness of head prediction was then examined. Since the drug is the head portion of the triple (Drug, TREATS, Disease), head prediction provides the answer to the question: “Which drug is the treatment for the disease?” The MRR head prediction score significantly increased from the HittER assessment score of 12.31, reaching a value of 21.81 in the findings of the head prediction. When compared to the highest baseline value of 5.33, the Hits@1 score shows a 126.83% increase in numbers. This result once more demonstrates that our suggested model outperformed the HittER model, which simply uses structural representations. Regarding the StAR inverse model, Table 5 reports the tail prediction results since the drug section is really in the inverse model's tail input (Disease, IS_TREATED_BY, Drug).

Table 5 The performance of the link prediction model for treatment-related relation is shown in the "head" column, which only displays results from head predictions, while the "both" column averages results from both head and tail predictions. Bold text indicates the top performance score. As the dataset triple direction is reversed, the StAR inverse result is based on the tail prediction.

Model	MRR(↑)		Hits@1(↑)		Hits@3(↑)		Hits@10(↑)	
	both	head	both	head	both	head	both	head
StAR	12.00	9.24	4.15	4.38	15.34	10.55	28.25	19.55
StAR inv	13.16	6.84	8.00	3.44	14.22	6.75	23.64	14.22
HittER	28.26	12.31	18.96	5.33	33.77	12.44	45.85	31.28
Ours	**31.06**	**21.81**	**22.10**	**12.09**	**36.97**	**22.39**	**46.15**	**32.58**

5.1.3. Description ablation study

This section aims to demonstrate how adding entity descriptions may enhance a model's capabilities. To compare the suggested model that makes use of description embeddings with a no-description version, we trained the textual model with and without entity descriptions for both the normal and inverse versions. Table 6 compares the MRR values for each setting. Overall, we saw a common tendency in all the models: when trained on data containing descriptions, all metric values rose. Such a finding demonstrates how including descriptions gives the model more meaningful data for forecasting. The MRR values of StAR dramatically increased from 11.69 to 13.32 and from 14.92 to 16.43 for the StAR inversed model after including the additional entity descriptions as input. It should be highlighted that the structural model and description-added textual model together achieved the greatest MRR and Hits@1 values of any model, at 27.19 and 18.18, respectively.

Table 6 Link prediction model comparison between those with (w/) and without (w/o) descriptions. The highest performance rating is highlighted in bold.

Model	MR (↓)	MRR(↑)	Hits@1(↑)	Hits@3(↑)	Hits@10(↑)	Hits@100(↑)
StAR w/o des	4602.98	11.69	4.23	15.29	25.54	44.27
StAR w/ des	**4203.58**	**13.32**	**6.73**	**15.63**	**27.01**	**45.46**
StAR inv w/o des	4801.23	14.92	8.87	17.32	26.84	44.39
StAR inv w/ des	**4650.22**	**16.43**	**10.10**	**19.04**	**28.58**	**45.91**
Ours w/o des	4509.87	26.24	17.32	32.07	42.01	49.61
Ours w/ des	**4502.62**	**27.19**	**18.18**	**33.23**	**43.17**	**50.11**

5.2. Synonym augmentation

In the main experiment, we only use the entities and their descriptions as the representation of the textual information. However, in real life drugs can also have multiple names used in literature. For example, Metformin can also be known as "1,1-Dimethylbiguanide", or "Dimethylbiguanid". Thus, we would like to test whether the performance of the model can be improved model performance even further. The entities with the semantic type of "Organic Chemical" and "Pharmacologic Substance" are chosen to search for their entities. The total number of entities of this type is 5,680 with only 965 found to have additional synonyms. The total number of synonyms is 3,929 terms. The augmented triples are created by every possible combination based on the existing triple. For example, for a triple "A TREATS B" if A has 5 total names and B has 3 total names (including the original entity name) the total number of possible triples quantity is 15. Thus, the newly augmented triples of this example are 14 triples without counting the original combination. We do this for all the triples in the training set and finally, the total number of augmented triples is 114,823 triples. Since synonyms are different words for the same entity node, they can only be used to apply to a model with a textual encoder. The result of the augmentation is shown in Table 7. The model with augmented triples only surpasses the Hits@1 value of the original StAR model. One

explanation for this is that the StAR is trained with more triples of a certain entity making the model always predicting the same entities frequently pushing the ranking to the highest top value. While starting from Hits@3 the model performance begins to drop significantly with a value of 37.53 in comparison to the Hits@100 value of 44.27 of the original StAR model. This may be the result of a triple without any synonym being overshadowed by other high-number synonym triples, thus, making the model not rank the correct triple properly. Moreover, most research papers has a standard naming for certain drugs. The additional synonyms are mostly just nicknames to the drugs and do not appear in literature. As the language model is trained on research papers, the new additional synonyms may not be fully learned during the pre-trained period compared to the standard term.

Table 7 Comparing textual encoder model with and without augmentation

Model	MR(↓)	MRR(↑)	Hits@1(↑)	Hits@3(↑)	Hits@10(↑)	Hits@100(↑)
StAR	**4602.98**	**11.69**	4.23	**15.29**	**25.54**	**44.27**
StAR with synonym augmented	6266.46	10.64	**6.15**	11.53	19.66	37.53

5.3. Filter negative sample experiment

The most general method of link prediction is to replace the head or tail part with all the entities that existed in the graph to create corrupted triples. However, the flaw of this method is that each entity has its semantic type like a drug or disease. Not all types are compatible with one another in forming a triple. For example, a drug cannot treat another drug. Therefore, replacing the triple without considering the type of entity could create negative triples that may not be possible at all in the first place. We performed two experiments on filtering these negative samples.

5.3.1. Re-evaluation with negative sample filtering

The first experiment is done to test the performance of the originally trained model on evaluation with filtering on negative sample setting. As described in section 2.4, the evaluation process is performed by corrupting the positive triple to create negative triples. However, in this experiment, only those triples that are compatible with the incomplete triple are counted in the ranking. Table 8 Illustrated the type of entities that could appear as the head part of the triple with “TREATS” relation and “Disease”.

Table 8 Schema of a triple with TREATS as the relation and "Disease or Syndrome" as the tail type

Head type	Relation	Tail type
Amino Acid, Peptide, or Protein	TREATS	Disease or Syndrome
Antibiotic	TREATS	Disease or Syndrome
Biologically Active Substance	TREATS	Disease or Syndrome
Biomedical or Dental Material	TREATS	Disease or Syndrome
Chemical Viewed Functionally	TREATS	Disease or Syndrome
Chemical Viewed Structurally	TREATS	Disease or Syndrome
Element, Ion, or Isotope	TREATS	Disease or Syndrome
Enzyme	TREATS	Disease or Syndrome
Food	TREATS	Disease or Syndrome
Gene or Genome	TREATS	Disease or Syndrome
Hazardous or Poisonous Substance	TREATS	Disease or Syndrome
Hormone	TREATS	Disease or Syndrome
Immunologic Factor	TREATS	Disease or Syndrome
Inorganic Chemical	TREATS	Disease or Syndrome
Indicator, Reagent, or Diagnostic Aid	TREATS	Disease or Syndrome
Molecular Function	TREATS	Disease or Syndrome

Nucleic Acid, Nucleoside, or Nucleotide	TREATS	Disease or Syndrome
Organic Chemical	TREATS	Disease or Syndrome
Pharmacologic Substance	TREATS	Disease or Syndrome
Receptor	TREATS	Disease or Syndrome
Vitamin	TREATS	Disease or Syndrome

The result of this experiment can be seen in Table 9. All the metrics value is better compared to the non-filtering method. This is not surprising as the ranking was done on a filtered list of negative samples. However, the leading value on the MRR and all the Hits@K values are very marginal which could be evidence that the model can already discriminate the non-compatible triple from the rest of the negative triple well. A huge increase is seen in the MR metrics. One possible explanation is that although the new evaluation setting does not affect the positive triple that is already on the top ranking much, the ranking greatly jumps up for the poorly ranked positive triple. However, since our work is more concerned about the top rank value, the standard method has been already an adequate evaluation method of the models.

Table 9 Comparing the result of non-filtering and negative sample filtering methods

Model	MR (↓)	MRR(↑)	Hits@1(↑)	Hits@3(↑)	Hits@10(↑)	Hits@100(↑)
Non-filtering method (standard)	4502.62	27.19	18.18	33.23	43.17	50.11
Negative samples filtering method	3470.53	27.59	18.52	33.67	43.75	50.25

5.3.2. Training and evaluating with filtering method

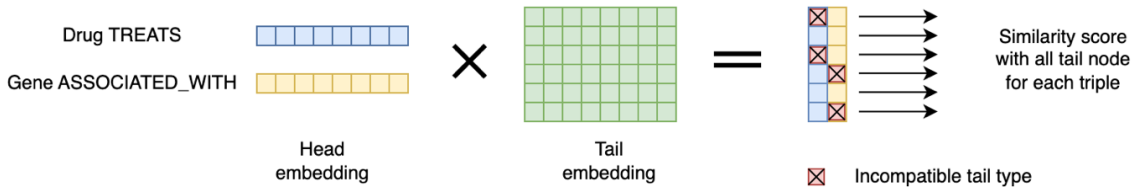


Figure 15 The similarity score calculation procedure of the filtering method

Figure 15 illustrated the similarity score calculation method between the head embedding and all the tail embeddings. Since not all tail types are compatible with the head part (including the relation), not all the similarity scores should be calculated during the loss calculation step. To discard the influence of the unwanted node score we replace their value with minus infinity. Since the probability of each head and tail node is achieved by using the softmax function the value of minus infinity will make the value of the probability becomes zero, thus canceling its effect. We re-trained the model with this loss calculation method and compared it to the standard training process. The result in Table 10 shows that standard training is still the best method to train the model. We assume that since the incompatible node similarity is still computed, it may result in instability during the loss calculation. Since we are working in batch, separately filtering, and calculating the similarity score with only the compatible nodes will need high computational power. We assume a better method compared to masking the unwanted score with minus infinity is needed.

Table 10 Comparing the training of standard and filtering method

Training method	MR(↓)	MRR(↑)	Hits@1(↑)	Hits@3(↑)	Hits@10(↑)	Hits@100(↑)
Standard training	3470.53	**0.2759**	**0.1852**	**0.3367**	**0.4375**	**0.5025**
Filter training	**3418.07**	0.2409	0.1498	0.2948	0.4227	0.4984

5.4. Different combination result

As seen in Figure 14, the embedding of the textual representation is only combined in the head part of the entity to enhance the performance of the structural model. To make sure our proposed method provided the best performance, a preliminary experiment is undertaken to test different combinations of the model: head-only, tail-only, and both head and tail enhancement. The result of this experiment is shown in Table 11. The result shows that head only is the best-performing combination method with the highest MRR score of 22.29. Although the head-only combination gives in to the MR metrics, the result of the rest of the metrics is far too low compared to the head-only combination, thus confirming that it is the best combination for the proposed model.

Table 11 Result of different enhancements on the triples part for the combined model

Model	MR(↓)	MRR(↑)	Hits@1(↑)	Hits@3(↑)	Hits@10(↑)	Hits@100(↑)
Head only	4789.49	**22.29**	**14.36**	**27.00**	**37.04**	**44.31**
Tail only	**3410.81**	6.38	2.27	7.81	13.00	28.57
Head and tail	3716.31	2.86	0.55	3.95	6.01	17.92

5.5. Discussion

From Table 4, we have seen that StAR performs well on MR metrics while failing on the others. Figure 16 shows the histogram of the entity ranking between three models: StAR, HittER, and our proposed model. We can see that StAR generally performs better than HittER and our model in the upper rank. While at the bins around 10,000 to 15,000, we can see the group of HittER-based model ranking which is stuck at the bottom. These clusters of low-ranked triples are the reason for the low performance of the MR score since this metric is calculated based on raw rank value.

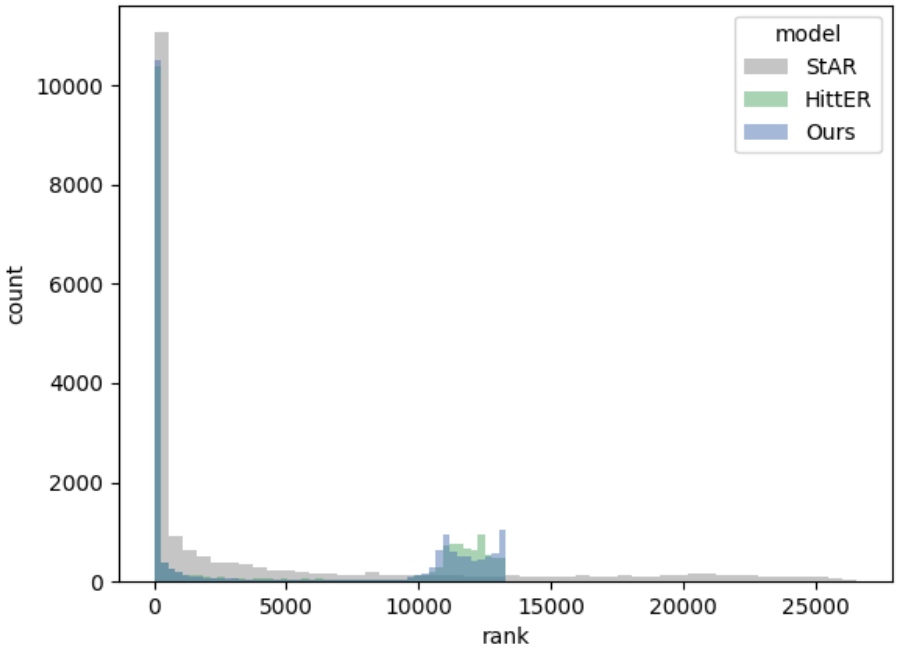


Figure 16 All ranking histogram of StAR and HittER

However, from Figure 17, in the top 100, our model and HittER model significantly outperforms the StAR model, especially in the first 2 bins. This is the reason why HittER-based models are better models for predicting entities at the top ranks while at the same time, many entities are gathered at the bottom which makes its overall ranking worse than StAR. Since our model used HittER as the backbone it exhibits similar results with the HittER model while also improving the ranking result. For the link prediction task, we are more interested in getting all the feasible drugs inside the top ranking as much as possible which is why a better performance in the top 100 could conclude that our proposed model is the better model for drug repurposing use case.

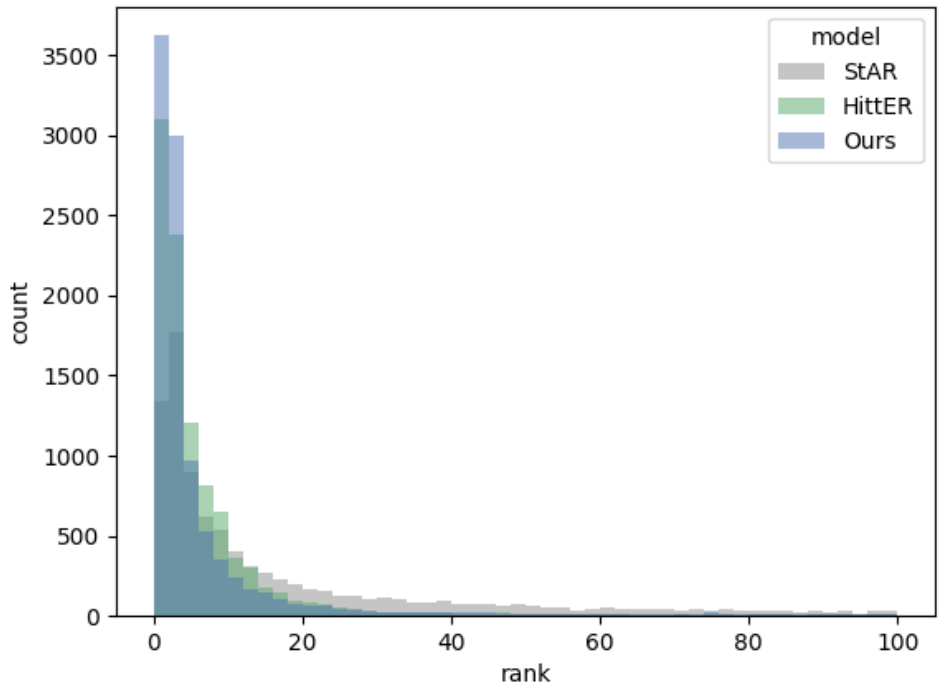


Figure 17 Top 100 ranking histogram

5.6. Type 2 diabetes drugs repurposed by the model

We obtained a drug list predicted by the model on (___, TREATS, Type 2 Diabetes) triple to demonstrate the model's capacity to acquire new medicines for disease treatment. To avoid undesirable unrelated medicine node types, we intentionally chose just the “Pharmacologic Substance” and “Organic Chemical” node types. In addition, as type 2 diabetes-related treatments were already present in the training, validation, and test sets, we filtered them out. Treatment that has already been shown to “TREATS” and “PREVENTS” interaction with “Type 2 Diabetes”, “Type 1 Diabetes”, and diabetes in general is not considered new treatment. Therefore, using the confidence scores of the drugs, we came up with a list of the top 10. We selected 5 medications with a high potential for repurposing by considering evidence from the literature and drug interactions in the dataset (refer to the table in each subsection).

The medicines associated with type 2 diabetes mostly emerge in the literature before the date of our training cut-off in the dataset. The drug list shows

that, even in situations when these connections are supported by the literature but absent from our dataset, our algorithm is still able to find the missing ties between medications and diseases. Despite the dataset's restrictions, we can successfully detect and extract relevant associations between medications and diseases by utilizing our model. This illustrates the usefulness of our method.

5.6.1. Triterpenes

This medication has the impact of accelerating biological processes in the heart and metabolism-related diseases [25]. Reducing postprandial glucose levels is reportedly a technique for managing type 2 diabetes. Alpha-glucosidases and alpha-amylases, which delay the absorption of carbohydrates in the intestine and lower the postprandial insulin level, can be inhibited to achieve this [26]. An interaction between terpenes and alpha-glucosidases (shown in Table 12) has been seen in one of the training triples in our dataset. This demonstrates that the model can infer other links in the network from existing triples to forecast new medications.

Table 12 Triterpenes

Head name	Relation	Tail name
tolbutamide	INTERACTS_WITH	Triterpenes
Triterpenes	AFFECTS	Diabetes
Triterpenes	COEXISTS_WITH	acarbose
Triterpenes	INHIBITS	Alpha-glucosidase
Triterpenes	INTERACTS_WITH	tolbutamide
acarbose	COEXISTS_WITH	Triterpenes
Triterpenes	AFFECTS	Carbohydrate Metabolism
Triterpenes	AFFECTS	cholesterol metabolism
Triterpenes	INTERACTS_WITH	Disulfides

5.6.2. Sho-saiko-to

Another intriguing discovery from our model is a herbal medication known as Sho-saiko-to. This herbal drug has been given orally to patients in Japan with chronic liver disease [27]. Our dataset reveals that this medication interacts substantially with tolbutamide, one of the medications used to treat type 2 diabetes [28], through many sorts of relations such as “INTERACTS_WITH”, “COEXISTS_WITH”, and “STIMULATES”. The interaction is illustrated in Table 13. The drug's ability to reduce blood glucose after 120 minutes of glucose loading was also confirmed by a study into a glucose tolerance test on a diabetic rat. A further indication of the drug's potential in lipid and mineral metabolism-related pathological conditions of diabetes mellitus is the drug's ability to lower cholesterol levels in the kidney's elastin-cholesterol fraction [29].

Table 13 Sho-saiko-to

Head name	Relation	Tail name
tolbutamide	COEXISTS_WITH	Sho-saiko-to
tolbutamide	INTERACTS_WITH	Sho-saiko-to
Sho-saiko-to	COEXISTS_WITH	tolbutamide
Sho-saiko-to	INHIBITS	tolbutamide
Sho-saiko-to	INTERACTS_WITH	tolbutamide
Sho-saiko-to	STIMULATES	tolbutamide

5.6.3. LY294002

The major enzyme that LY294002 inhibits is phosphatidylinositol 3-kinase (PI3K) [30]. This raises the potential that PI3Ks may be implicated in the onset of diabetes mellitus because PI3Ks are essential for managing glucose levels. It has been shown that inhibiting PI3K G protects against the onset of diabetes while activating PI3K A protects against heart failure brought on by diabetes [31]. Additionally, in Table 14, this medication interacts with several diabetic medications,

including “Glyburide”, “Metformin”, and “Exenatide”. We suppose that because of these relationships, the model gives this medicine a high rating.

Table 14 LY 294002

Head name	Relation	Tail name
glyburide	INTERACTS_WITH	LY 294002
Metformin	INTERACTS_WITH	LY 294002
exenatide	INTERACTS_WITH	LY 294002
exenatide	STIMULATES	LY 294002
LY 294002	COEXISTS_WITH	insulin glargine
LY 294002	INTERACTS_WITH	glyburide
LY 294002	INTERACTS_WITH	exenatide
LY 294002	INTERACTS_WITH	rosiglitazone
LY 294002	STIMULATES	liraglutide
LY 294002	INTERACTS_WITH	Metformin
rosiglitazone	INTERACTS_WITH	LY 294002
insulin glargine	COEXISTS_WITH	LY 294002

5.6.4. Clomiphene Citrate

This medication works by increasing the production of the hormones that aid in the formation and release of a mature egg, which is how it treats female infertility [32]. Even while just a few researchers have studied how clomiphene citrate affects glucose metabolism, they have all found favorable results. Clomiphene citrate improved insulin and glucose levels in obese dysmetabolic individuals with low testosterone levels, [33] suggesting its potential use in participation. As a result, Clomiphene Citrate has a connection to both glucose and fat, which in turn has a direct bearing on type 2 diabetes. As shown in Table 15, the drug has multiple relationships with the prominent type 2 diabetes drug, Metformin.

Table 15 Clomiphene Citrate

Head name	Relation	Tail name
bromocriptine	COEXISTS_WITH	Clomiphene Citrate
Metformin	COEXISTS_WITH	Clomiphene Citrate
Metformin	INTERACTS_WITH	Clomiphene Citrate
rosiglitazone	STIMULATES	Clomiphene Citrate
Clomiphene Citrate	COEXISTS_WITH	bromocriptine
Clomiphene Citrate	COEXISTS_WITH	Metformin
Clomiphene Citrate	INTERACTS_WITH	Metformin

5.6.5. Mitogen-activated protein kinase inhibitors (MAPK Inhibitors)

The two primary pathogenic processes that lead to the development of type 2 diabetes are insulin resistance and beta cell dysfunction. According to one research on p38 MAPK (p38), this protein plays a part in stress and inflammatory responses [34]. The p38 pathway is activated, and this results in ERS and inflammatory responses, which kill beta cells. The same study's findings showed that suppressing P38 MAPK might, at least in part, by reducing cell death, lower blood sugar levels and improve cell function [34]. From the training set, MAPK inhibitors frequently coexist and interact with prominent type 2 diabetes medications like exenatide and metformin, respectively. As seen in Table 16, although there are only a few interactions existing in the dataset, the model can rank this drug highly.

Table 16 Lipoxins

Head name	Relation	Tail name
pioglitazone	STIMULATES	Lipoxins
rosiglitazone	STIMULATES	Lipoxins
Lipoxins	TREATS	Macular edema due to diabetes mellitus
Lipoxins	AFFECTS	Diabetes

CHAPTER 6

Conclusion

In this research, we introduce a novel link prediction model for type 2 diabetes drug repurposing based on a modern deep learning architecture known as “transformer”. By combining the structural information from each entity's neighboring nodes and the textual information from its name and description, a graph representation of the entity is created. We believe that this is the first model that incorporates both elements of the link prediction model of transformer type. The textual encoder is first trained over a pre-trained language model. The textual property of the entity and relation is then embedded using the model. A second transformer model is trained to create the structural representation's embedding. Through fully connected layers, the two different types of representations are combined and fused. Lastly, the feature provided by the encoder is used to determine the triples' scores. The rating of the true triple can be obtained to evaluate the model after corrupted triples are sorted in descending order using the score.

Overall, our findings showed that our suggested design performed better than models that just used one type of embedding technique. On tasks relevant to treatment, the mean rank reciprocal value nearly doubles from 12.31 to 21.81, indicating a significant improvement over the best baseline model. Similar enormous number jumps can be seen in the Hits@K measures, especially at the low K value, as evidenced by the Hits@1 metrics' increase of 126.33%. These findings support the model's capacity to accurately forecast the right drug's high ranking. Finally, we found many medications that have a promising potential for repurposing from the top of the sorted list predicted by the algorithm. Our research presents a promising strategy for type 2 diabetes medicine repurposing and has the potential to significantly improve patient outcomes and healthcare expenditures. Although our results mark a significant first step in drug repurposing, additional research and screening are

required to confirm the viability of the suggested approach. The non-explainable nature of our model further emphasizes the necessity for further research on explainable models, which would offer insightful data to enhance medication shortlist decision-making.

REFERENCES

1. *Diabetes*. 2022 11 January 2023]; Available from: https://www.who.int/news-room/fact-sheets/detail/diabetes.
2. Cherney, K. *A Complete List of Diabetes Medications*. 2022 2023, January 10]; Available from: https://www.healthline.com/health/diabetes/medications-list.
3. Zhang, R., et al., *Drug repurposing for COVID-19 via knowledge graph completion.* Journal of Biomedical Informatics, 2021. **115**: p. 103696.
4. Vaswani, A., et al., *Attention is all you need.* Advances in neural information processing systems, 2017. **30**.
5. Devlin, J., et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. Minneapolis, Minnesota: Association for Computational Linguistics.
6. Yao, L., C. Mao, and Y. Luo, *KG-BERT: BERT for knowledge graph completion.* arXiv preprint arXiv:1909.03193, 2019.
7. Wang, B., et al. *Structure-augmented text representation learning for efficient knowledge graph completion*. in *Proceedings of the Web Conference 2021*. 2021.
8. Chen, S., et al. *HittER: Hierarchical Transformers for Knowledge Graph Embeddings*. 2021. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.
9. Pushpakom, S., et al., *Drug repurposing: progress, challenges and recommendations.* Nature Reviews Drug Discovery, 2019. **18**(1): p. 41-58.
10. Dosovitskiy, A., et al., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021.
11. Zheng, D., et al. *Dgl-ke: Training knowledge graph embeddings at scale*. in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2020.
12. Lin, Y., et al. *Learning entity and relation embeddings for knowledge graph completion*. in *Twenty-ninth AAAI conference on artificial intelligence*. 2015.

13. Yang, B., et al., *Embedding Entities and Relations for Learning and Inference in Knowledge Bases*, in *International Conference on Learning Representations (ICLR)* 2015.
14. Trouillon, T., et al. *Complex embeddings for simple link prediction*. in *International conference on machine learning*. 2016. PMLR.
15. Nickel, M., V. Tresp, and H.-P. Kriegel. *A three-way model for collective learning on multi-relational data*. in *Icml*. 2011.
16. Sun, Z., et al., *Rotate: Knowledge graph embedding by relational rotation in complex space.* arXiv preprint arXiv:1902.10197, 2019.
17. Welling, M. and T.N. Kipf. *Semi-supervised classification with graph convolutional networks*. in *J. International Conference on Learning Representations (ICLR 2017)*. 2016.
18. Wu, Z., et al., *A comprehensive survey on graph neural networks.* IEEE transactions on neural networks and learning systems, 2020. **32**(1): p. 4-24.
19. Nathani, D., et al. *Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs*. 2019. Florence, Italy: Association for Computational Linguistics.
20. Kilicoglu, H., et al., *SemMedDB: a PubMed-scale repository of biomedical semantic predications.* Bioinformatics, 2012. **28**(23): p. 3158-3160.
21. Kazemi, S.M. and D. Poole, *Simple embedding for link prediction in knowledge graphs.* Advances in neural information processing systems, 2018. **31**.
22. Lacroix, T., N. Usunier, and G. Obozinski. *Canonical tensor decomposition for knowledge base completion*. in *International Conference on Machine Learning*. 2018. PMLR.
23. Lee, J., et al., *BioBERT: a pre-trained biomedical language representation model for biomedical text mining.* Bioinformatics, 2020. **36**(4): p. 1234-1240.
24. Wolf, T., et al. *Transformers: State-of-the-Art Natural Language Processing*. 2020. Online: Association for Computational Linguistics.
25. Teng, H., et al., *Dietary triterpenes in the treatment of type 2 diabetes: To date.* Trends in Food Science & Technology, 2018. **72**: p. 34-44.
26. Nazaruk, J. and M. Borzym-Kluczyk, *The role of triterpenes in the management*

of diabetes mellitus and its complications. Phytochem Rev, 2015. **14**(4): p. 675-690.

27. Sakaida, I., et al., *Herbal medicine Sho-saiko-to (TJ-9) prevent liver fibrosis and enzyme-altered lesions in rat liver cirrhosis induced by a choline-deficient l-amino acid-defined diet.* Journal of Hepatology, 1998. **28**(2): p. 298-306.
28. Furman, B.L., *Tolbutamide*, in *xPharm: The Comprehensive Pharmacology Reference*, S.J. Enna and D.B. Bylund, Editors. 2007, Elsevier. p. 1-4.
29. Goto, M., et al., *[Effects of traditional Chinese medicines (dai-saiko-to, sho-saiko-to and hachimi-zio-gan) on* spontaneously *diabetic rat (WBN/Kob) with experimentally induced lipid and mineral disorders].* Nihon Yakurigaku Zasshi, 1992. **100**(4): p. 353-8.
30. Denny, W.A. and G.W. Rewcastle, *Chapter 15 - Inhibitors of the Phosphatidylinositol 3-Kinase Pathway*, in *Cancer Drug Design and Discovery (Second Edition)*, S. Neidle, Editor. 2014, Academic Press. p. 449-478.
31. Maffei, A., G. Lembo, and D. Carnevale, *PI3Kinases in Diabetes Mellitus and Its Related Complications.* Int J Mol Sci, 2018. **19**(12).
32. Vardanyan, R.S. and V.J. Hruby, *28 - Female Sex Hormones*, in *Synthesis of Essential Drugs*, R.S. Vardanyan and V.J. Hruby, Editors. 2006, Elsevier. p. 365-379.
33. Gambineri, A. and C. Pelusi, *Sex hormones, obesity and type 2 diabetes: is there a link?* Endocr Connect, 2019. **8**(1): p. R1-r9.
34. Wei, X., et al., *Inhibition of p38 mitogen-activated protein kinase exerts a* hypoglycemic effect by improving β cell function via inhibition of β cell *apoptosis in db/db mice.* Journal of Enzyme Inhibition and Medicinal Chemistry, 2018. **33**(1): p. 1494-1500.
35. *Diabetes Drugs*. 2019 10 June 2022; Available from: https://www.diabetes.co.uk/Diabetes-drugs.html.
36. Bodenreider, O., *The Unified Medical Language System (UMLS): integrating biomedical terminology.* Nucleic Acids Res, 2004. **32**(Database issue): p. D267-70.

APPENDIX A

DRUG LIST FOR DATA SCRAPING

In this appendix, we list all the drugs that are used during the process of scraping triples from SemMedDB. The list provided by [2, 35]. Although included in this list, any entities that could not be found in UMLS Metathesaurus [36] are discarded from the final drug scraping list.

Table 17 Drug Names used in data scraping.

Diabetes Type	Drug Name
Type 1 Diabetes Drugs	Humulin
	Novolin
	NovoLog
	FlexPen
	Fiasp
	Apidra
	Humalog
	Humulin N
	Novolin N
	Tresiba
	Levemir
	Lantus
	Toujeo
	NovoLog Mix 70/30
	Humalog Mix 75/25
	Humalog Mix 50/50
	Humulin 70/30
	Novolin 70/30
	Ryzodeg
	Pramlintide
	SymlinPen

Type 2 Diabetes Drugs	acarbose
	miglitol
	metformin
	Kazano
	Invokamet
	Xigduo XR
	Synjardy
	Glucovance
	Jentadueto
	Actoplus
	PrandiMet
	Avandamet
	Kombiglyze XR
	Janumet
	Bromocriptine
	alogliptin
	alogliptin-metformin
	alogliptin-pioglitazone
	linagliptin
	linagliptin-empagliflozin
	linagliptin-metformin
	saxagliptin
	saxagliptin-metformin
	sitagliptin
	sitagliptin-metformin
	sitagliptin
	Vildagliptin
	albiglutide
	dulaglutide
	exenatide
	exenatide extended-release
	liraglutide
	semaglutide

	Lixisenatide
	nateglinide
	repaglinide
	repaglinide-metformin
	dapagliflozin
	dapagliflozin-metformin
	canagliflozin
	canagliflozin-metformin
	empagliflozin
	empagliflozin-linagliptin
	empagliflozin-metformin
	ertugliflozin
	glimepiride
	glimepiride-pioglitazone
	glimepiride-rosiglitazone
	gliclazide
	glipizide
	glipizide-metformin
	glyburide
	glyburide-metformin
	chlorpropamide
	tolazamide
	tolbutamide
	Glibenclamide
	Gliquidone
	Glyclopyramide
	rosiglitazone
	rosiglitazone-glimepiride
	rosiglitazone-metformin
	pioglitazone
	pioglitazone-alogliptin
	pioglitazone-glimepiride
	pioglitazone-metformin

VITA

NAME	Sothornin Mam
DATE OF BIRTH	18 October 1998
PLACE OF BIRTH	Kandal, Cambodia
INSTITUTIONS ATTENDED	Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University
HOME ADDRESS	No.215, St.113, Prek Samrong Village, Sangkat Takhmau, Takhmau City, Kandal Province, Cambodia
PUBLICATION	Mam, S., Wichadakul, D., & Vateekul, P. (2023). Drug Repurposing for Type 2 Diabetes Using Combined Textual and Structural Graph Representation Based on Transformer. IEEE Access, 11, 65711-65724. https://doi.org/10.1109/ACCESS.2023.3289863
AWARD RECEIVED	72nd Anniversary of His Majesty King Bhumibol Adulyadej and International Graduate Students Scholarship Chulalongkorn University
	Royal Scholarship under Her Royal Highness Princess Maha Chakri Sirindhorn Education Project to the Kingdom of Cambodia For 2017 Prince of Songkla University