2019

Relation Prediction over Biomedical Knowledge Bases for Drug Repositioning

Mehmet Bakal
University of Kentucky, mgokhanbakal@hotmail.com

Digital Object Identifier: https://doi.org/10.13023/etd.2019.402

Mehmet Bakal, Student

Dr. Ramakanth Kavuluru, Major Professor

Dr. Mirosław Truszczyński, Director of Graduate Studies

Relation Prediction over Biomedical Knowledge Bases for Drug Repositioning

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
in the College of Engineering at the
University of Kentucky

By
Mehmet Gokhan Bakal
Lexington, Kentucky

Co-Director: Dr. Ramakanth Kavuluru, Associate Professor of Biomedical Informatics
Co-Director: Dr. Zongming Fei, Professor of Computer Science
Lexington, Kentucky 2019

ABSTRACT OF DISSERTATION

Relation Prediction over Biomedical Knowledge Bases for Drug Repositioning

Identifying new potential treatment options for medical conditions that cause human disease burden is a central task of biomedical research. Since all candidate drugs cannot be tested with animal and clinical trials, in vitro approaches are first attempted to identify promising candidates. Likewise, identifying other essential relations (e.g., causation, prevention) between biomedical entities is also critical to understand biomedical processes. Hence, it is crucial to develop automated relation prediction systems that can yield plausible biomedical relations to expedite the discovery process. In this dissertation, we demonstrate three approaches to predict treatment relations between biomedical entities for the drug repositioning task using existing biomedical knowledge bases. Our approaches can be broadly labeled as link prediction or knowledge base completion in computer science literature. Specifically, first we investigate the predictive power of graph paths connecting entities in the publicly available biomedical knowledge base, SemMedDB (the entities and relations constitute a large knowledge graph as a whole). To that end, we build logistic regression models utilizing semantic graph pattern features extracted from the SemMedDB to predict treatment and causative relations in Unified Medical Language System (UMLS) Metathesaurus. Second, we study matrix and tensor factorization algorithms for predicting drug repositioning pairs in `repoDB`, a general purpose gold standard database of approved and failed drug–disease indications. The idea here is to predict `repoDB` pairs by approximating the given input matrix/tensor structure where the value of a cell represents the existence of a relation coming from SemMedDB and UMLS knowledge bases. The essential goal is to predict the test pairs that have a blank cell in the input matrix/tensor based on the shared biomedical context among existing non-blank cells. Our final approach involves graph convolutional neural networks where entities and relation types are embedded in a vector space involving neighborhood information. Basically, we minimize an objective function to guide our model to concept/relation embeddings such that distance scores for positive relation pairs are lower than those for the negative ones. Overall, our results demonstrate that recent link prediction methods applied to automatically curated, and hence imprecise, knowledge bases can nevertheless result in high accuracy drug candidate prediction with appropriate configuration of both the methods and datasets used.

KEYWORDS: Biomedical Relation Prediction, Machine Learning, Computational Drug Repositioning, Tensor Factorization, Graph Convolutional Neural Networks

Author's signature: Mehmet Gokhan Bakal

Date: October 18, 2019

Relation Prediction over Biomedical Knowledge Bases for Drug Repositioning

By
Mehmet Gokhan Bakal

Co-Director of Dissertation: Ramakanth Kavuluru

Co-Director of Dissertation: Zongming Fei

Director of Graduate Studies: Mirosław Truszczyński

Date: October 18, 2019

To my parents, my wife, and son.

ACKNOWLEDGMENTS

First of all, I wish to express sincere appreciation to Dr. Ramakanth Kavuluru for the five long years as my advisor. His support and advice have guided me through the most difficult moments of my research. Then, I would like to thank my committee members, Dr. Jerzy J. Jaromczyk, Dr. Zongming Fei, Dr. Penni E. Black, and Dr. Kenneth Kubota for their time and valuable suggestions. Along with other faculty members, I would like to express my sincere gratitude to Dr. Mirosław Truszczyński, director of graduate studies at the department of computer science, for his guidance throughout my graduate program. Also, I gratefully acknowledge the Ministry of National Education, Republic of Turkey, for providing financial support with a full scholarship for my doctoral studies. Finally, I would also like to thank my parents, my twin brother and my wife for their moral support. Their valuable encouragement helped me a lot to focus on my graduation without any distraction.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Biomedical Knowledge Discovery

Manual analysis and exploration of medical data is increasingly becoming non-trivial given the exponential growth in numbers and sizes of such datasets owing to advances in data science and general quality of EMRs. Therefore, there is a lack of automatic processing systems for not only analyzing the data but also discovering/extracting any previously unknown relationships between biomedical entities such as treatment-causative relations or drug-drug interactions (Cheng et al., 2014; Zhang et al., 2014c). In order to automate manual operations on a vast amount of rapidly growing biomedical data, many researchers employ computational methods including machine learning (ML) and natural language processing (NLP) as well as graph modeling techniques (Cameron et al., 2015b; Wilkowski et al., 2011; Workman et al., 2016; Zhang et al., 2011). This process of using automated methods to elicit new information is generally termed as knowledge discovery.

In biomedical domain, knowledge can be defined as meaningful interpretation or information that can be gleaned from a medical dataset, available either in a structured form or textual form. In general sense, this could mean named entities of interest to users, such as names of drugs, diseases, and procedures. High level knowledge typically involves interactions between these extracted biomedical named entities. For example, there is typically a treatment relation between a drug and a disease. Here, these interactions are generally called relations that connect a subject entity (*drug*) with an object entity (*disease*) through a predicate (*treats*). Beyond just named entities, a set of meaningful relations extracted from a dataset can also be construed as a more specific kind of knowledge. However, indirect or implicit relations might exist and can be obtained by putting together several known relations as a sequence where the entities at either end of the sequence are now seen as participating in a new relation. But this can only happen when the nature of entities and predicates along the sequence is meaningful to derive this new connection. For example, consider this sequence of two relations with *stimulates* and *treats* predicates in that order below:

$$\text{Mercaptopurine} \xrightarrow[Stimulates]{} \text{Cytarabine triphosphate} \xrightarrow[Treats]{} \text{Leukemia}$$

From the two constituent relations taken in this specific order, we can now see a potential new relation: (Mercaptopurine - *treats* - Leukemia). In fact, this is known to be a known relation between Mercaptopurine and Leukemia. Hence, this information is called implicit knowledge as it is not explicitly expressed in medical textual narratives or structured data sources.

Many simple paths or sequences of relations clearly do not all lead to new undiscovered relations. Even when they do, they may not sometimes be interpretable in a biomedical sense due to missing additional context. There are cases where a compact subgraph connecting a pair of entities is essential in inferring a new implicit relation (Cameron et al., 2015b). Hence, the process of discovery might also involve identifying interesting subgraphs that convey a richer and more comprehensive overview of the new relation.

At a high level, for the purposes of this thesis, knowledge discovery entails discovery of new binary relations (e.g., treatment, causative, preventative) connecting pairs of biomedical entities based on existing relation databases that are either manually curated (not uncommon in medicine, such as the unified medical language system (UMLS) Metathesaurus (National Library of Medicine, 2009)) or automatically curated (through NLP methods, such as the Semantic Medline Database (National Library of Medicine, 2016)). This discovery is carried out through a prediction process typically modeled by machine learning methods. A particular focus is on the drug repositioning problem, where based on existing knowledge about already approved drugs or drugs that were found to be safe in humans, new use-cases of drugs are identified for specific new conditions. That is, the repositioning task is a special case of knowledge discovery where one predicts treatment relations from existing treatment relations or a much larger set of relations involving other predicates. Given prediction is inherently prone to errors, hypothesis generation is sometimes employed as a more appropriate name sometimes employed by researchers to discuss computational methods for knowledge discovery.

1.1 Thesis Statement and Summary

Discovering (and as a first step, hypothesizing) new biomedical relations is a core task in advancing biomedical research. Specifically, identifying new viable drug candidates for diseases is vital due to the excessive time and financial cost burden of a traditional drug development pipeline. Our hypothesis is that advances in link prediction methods that are carefully configured and applied to existing knowledge bases (curated either manually or in an automated manner) can aid in building high accuracy models for relation prediction in general and specifically, for drug repositioning. We verify this hypothesis using three different efforts.

First, we employ logistic regression models that use semantic graph patterns connecting pairs of entities to this end and predict treatment and causative relations. Although effective, extracting pairs of *all* paths (from which patterns are derived) of lengths $\leq k$ quickly becomes intractable even for small k (e.g., $k \geq 4$) for moderately dense graphs. Next, we explore and show that matrix completion through low-rank approximation via nonnegative matrix factorization (NMF) can be a simpler and more efficient method that can be configured for the specific task of drug repositioning using the UMLS and SemMedDB knowledge bases. Although much faster, matrix completion methods cannot identify new viable drug–disease pairs for drugs for which there is not even a single prior known positive treatment relation (with some other condition). A drug that is not FDA approved for any condition may still be a viable candidate for other conditions if the clinical trails that failed did not show any adverse affects in humans (besides simply being ineffective for the conditions it was considered for approval). To address these scenarios, in our final effort, we use tensor factorization (TF) and the more recent graph convolutional networks (GCNs) with neural embeddings of the entities and relations ($\in \mathbb{R}^d$) as more effective alternatives for drug candidate prediction (even for drugs without a single prior known treatment relation). We demonstrate that these methods improve recall by trading off precision in comparison with matrix completion methods. At a high level the intuition for this effect is the ability of TF and GCN methods to exploit multi-hop connections between candidate drugs and diseases to predict new relations. Owing to a potential complementary evidences captured by each of our methods, an ensemble model that combines NMF, TF, and GCN predictions performs better than all three constituent models in terms of F1-score. For the second and third efforts, we use the `repoDB` (Brown and Patel, 2017) dataset that was explicitly created to evaluate computational drug repositioning efforts, and includes both FDA approved drugs and failed indication from ClinicalTrials.gov.

1.2 Organization

The remaining chapters in this dissertation are organized as follows:

Chapter 2 presents relevant prior efforts on biomedical relation prediction and knowledge discovery. The notations used throughout this manuscript and the evaluation metrics used are also introduced here. This chapter briefly covers the basics of supervised machine learning and the fundamental definitions of the algorithms utilized in each chapter.

Chapter 3 introduces a novel and intuitive approach which exploits semantic graph patterns as features to predict treatment and causative relations between any given pair of biomedical entities. We build logistic regression (LR) and decision tree (DT) models with graph pattern features. We also provide the details about the potential of graph patterns in terms of coverage and utility of highly discriminative patterns identified through coefficients of our best LR model.

Chapter 4 details matrix completion through NMF for drug repositioning. First, we describe how we applied the factorization approach by curating treatment relations from SemMedDB and UMLS Metathesaurus to form the input matrix for the factorization process. Then, we discuss the NMF 'scores' for the `repoDB` pairs and predictive performance of a few NMF models. In addition, we present the influence of exploiting chronological information of input treatment relations (when they were first reported in literature) on the NMF scores of the drugs in approved treatments in `repoDB` test set.

Chapter 5 describes tensor factorization and knowledge graph embeddings with the GCN methods for classifying drug repositioning test pairs. We details the configurations of our models along with their performance scores compared with each other for the `repoDB` test pairs. We also present a simple ensemble model based on majority voting built with the best NMF, TF, and GCN configurations and evaluate its performance. Finally, we conduct error analyses and evaluations for the false positive and false negative predictions in collaboration with clinicians.

Chapter 6 summarizes this dissertation with the results and contributions from different experiments conducted in all other chapters. Besides this, we discuss several advantages and disadvantages of each method as well as the limitations and the future directions.

1.3 Related Publications

This dissertation contains material previously published in the following papers:

- **Gokhan Bakal** and Ramakanth Kavuluru. Predicting Treatment Relations with Semantic Patterns over Biomedical Knowledge Graphs. In the Proceeding of the International Conference on Mining Intelligence and Knowledge Exploration (MIKE) 2015 (pp. 586-596). [Chapter 3]

- **Gokhan Bakal**, Pretham Talari, Elijah V. Kakani, and Ramakanth Kavuluru. Exploiting Semantic Patterns over Biomedical Knowledge Graphs for Predicting Treatment and Causative Relations. In: Journal of Biomedical Informatics, 82, 2018 (pp. 189-199). [Chapter 3]

- **Gokhan Bakal**, Halil Kilicoglu, and Ramakanth Kavuluru. Non-Negative Matrix Factorization for Drug Repositioning: Experiments with the repoDB Dataset. (To appear in the Proceedings of the AMIA 2019 Annual Symposium). [Chapter 4]

The papers which are either submitted or under preparation are below:

- **Gokhan Bakal**, Romil Chadha, Tushi Singh, and Ramakanth Kavuluru. Predicting Drug Repositioning Pairs with Tensor Factorization and Graph Convolutional Networks. (Under preparation to be submitted to Journal of Biomedical Informatics or the Journal of American Medical Informatics Association).

Chapter 2 Related Work and Background

Given the exponential growth (Lu, 2011) of scientific literature, it is unrealistic to manually review all articles published on a given topic. Therefore, natural language processing (NLP) techniques have been increasingly used to *extract* biomedical relations from free text documents. For instance, the treatment relation example discussed earlier in this section may be extracted from the sentence – "*We conclude that Tamoxifen therapy is more effective for early stage breast cancer patients.*" However, NLP extractions are essentially based on evidence present in particular sentences and are prone to two types of errors. First, the NLP techniques themselves might not be foolproof and second the evidence found in a particular sentence might be circumstantial and not something that is universally accepted. Nevertheless, extraction of the same relation from multiple sentences might be indicative of the strength of the relation if it is being reported by multiple research projects.

Binary relations are typically encoded as (subject, predicate, object) triples and can be obtained from (1). well known expert curated knowledge bases, (2). applying NLP techniques to free text from literature, or (3). employing global lexico-syntactic pattern based methods. Due to excessive time consumption involved in manual curation, knowledge bases are generally not complete/exhaustive (Xu et al., 2013). NLP approaches can be used to extract relations from particular sentences using the linguistic structure of a sentence (syntactic/dependency parse) especially involving the spans of named entities that occur in it (Abacha and Zweigenbaum, 2011; Fundel et al., 2007; Kavuluru et al., 2012; Kim et al., 2015; Segura-Bedmar et al., 2011). Even though such systems are popular for relation extraction, they are error prone and might result in extraction of coincidental outcomes that cannot be considered general knowledge. Furthermore, implicit relations that are not necessarily asserted in a sentence cannot be obtained through such approaches. However, NLP extractions can be used as a basis to develop more advanced techniques that aim toward a global relation prediction modeling paradigm. This process of distilling the literature and gleaning actionable information that drastically reduces researcher efforts in dealing with information explosion is termed as literature-based discovery (LBD).

An alternative approach for the LBD is distant supervision (Mintz et al., 2009) (also called "weak supervision") when there are many predicates and manual labeling of sentences with relations is impractical. In this approach, pairs of entities, which come from a high quality knowledge base, are known to participate in specific relations and are used to search literature to identify sentences that contain both of them. Such sentences are used as training instances for the corresponding predicates to learn lexico-syntactic patterns that could be used as features in supervised models or in ranking new relations using unsupervised approaches (Xu et al., 2009). Although distant supervision offers a convenient approach to overcome the labeled data scarcity issue, a disadvantage is that existence of a pair of entities in a sentence does not directly mean that the sentence is expressing the particular relation existing in the knowledge base. Another important disadvantage is that the knowledge base could be incomplete and hence **negative example pairs (those that do not participate in a relation in the knowledge base) may not be true negatives**. Even though these disadvantages were comparatively mentioned and obviated by some other approaches (Riedel et al., 2010; Ritter et al., 2013; Surdeanu et al., 2012; Xu et al., 2013), few researchers have addressed these issues especially in biomedicine. In our first effort in Chapter 3, we propose an approach that is very different from these existing popular methods by relying on the graph path patterns extracted from a large graph of extracted relations using NLP approaches. As such, our methodology is not "extracting" a relation form a particular sentence, but is rather "predicting" a relation based on the implicit connections between the corresponding entities.

Similar to the relation extraction studies on the scientific literature, drug repositioning efforts have also gained an influential role against the traditional drug development methods. More interestingly, computational drug repositioning (CDR) studies help clinical researchers to shorten the drug discovery process by exposing more plausible candidate indications for the approved drugs. Since the scientific literature of treatment data has been exponentially growing, many computational approaches have been employed to explore new indications for the drugs (Li et al., 2016). Hence, the similarities of major characteristics of drugs such as chemical structures, molecular level activities, and side effects are the essential research directions for discovering new indications. Beyond the similarity based efforts, exploiting ontologies (Zhu et al., 2014) and drug-target networks (Li and Lu, 2012) as well as utilizing available large-scale genomic data sources (Dudley et al., 2011) are the other recent studies.

Unlike the CDR efforts mentioned above, we exploit matrix and tensor factorization and graph convolutional neural networks (GCN) approaches to predict drug repositioning pairs in `repoDB`. Factorization based algorithms are successfully applied to diagnostics of test results, analysis of genomic data, and completion of the electronic health records in biomedical domain (Luo et al., 2016; Roy et al., 2014; Wang et al., 2015). The elegance here is that the learned low dimensional latent embeddings capture the underlying patterns among the matrix/tensor entries for relation prediction and other various tasks such as document clustering and recommendation systems.

On the other hand, using neural embeddings of the nodes and edges in a knowledge graph is extensively studied for the knowledge graph completion task (Nickel et al., 2011; Yang et al., 2014; Wang et al., 2014; Ji et al., 2015; Trouillon et al., 2016). Similarly, GCN techniques are also utilized for several tasks such as learning molecular fingerprints for drug efficacy and predicting relations with muti-relational data (Duvenaud et al., 2015; Schlichtkrull et al., 2018). To that end, we utilize the embeddings of the semantic predications from SemMedDB and UMLS graphs. Besides, we also exploit the GCN algorithm which enhances the entity embeddings by considering the neighborhood information. Ultimately, we build models exploiting matrix/tensor factorization and GCN with SemMedDB and UMLS Metathesaurus to predict approved and failed drug-disease pairs in `repoDB`. Moreover, we demonstrate that the factorization and GCN models are both efficient on the relation prediction task while resulting in satisfactory predictive performance.

2.1 Notation

In this dissertation, vectors are represented as bold lower case letters (e.g. $\mathbf{x}$**,** $\mathbf{w}$ and $\mathbf{z}$); matrices are denoted as bold upper case letters (e.g. $\mathbf{X}$**,** $\mathbf{W}$ and $\mathbf{H}$) and high dimensional tensors are represented as upper case letters with Euler script font (e.g. $\mathcal{X}, \mathcal{T}$ and $\mathcal{K}$). Subscripts are used to differentiate different elements and vectors (e.g. $\mathbf{x_i}, \mathbf{w_i}$ and $\mathbf{X_i}, \mathbf{W_i}$). Superscripts are used to represent the value of the recent iteration/case for an argument (e.g. $\mathbf{x}^{l+1}$).

2.2 Supervised Machine Learning

The aim of supervised learning is to learn from already labeled training examples and predict labels for new unseen instances. In this dissertation, we focus on binary classification, where the main goal is to have machine learned models, which can generalize examples from training dataset, to assign a label, $y \in \{0, 1\}$, to every example in test dataset. Each example is represented by a n dimensional feature vector $\mathbf{x} = (x_1, .., x_n)$, where every feature x_i denotes unique predictive information. In order to generate this feature representation, examples are required to be transformed from the input space (e.g. frequencies of semantic patterns that exist between biomedical entities) into the n dimensional feature space. For supervised learning approaches (we utilized one of them; logistic regression in Chapter 3), the learning algorithm uses m labeled example pairs, $\{(x_1, y_1), .., (x_m, y_m)\}$ where y_i is the related label of feature vector x_i for each example i. Afterwards, the algorithm learns a statistical model on training dataset, which can be utilized to predict class labels on a test example with the same n dimensional feature vector. Some of the most widely used algorithms in supervised classification are naïve bayes, decision trees, k-nearest neighbors, logistic regression, support vector machines (SVM), and artificial neural networks.

2.2.1 Matrix & Tensor Factorization Techniques

Low-rank approximations of matrices and tensors via factorizations (or decompositions) play a major role in exploiting the data and extracting latent components for several applications such as text mining (Kuang et al., 2015), computer vision (Koudelka and Dorsey, 2016), and financial data analysis (Fréin et al., 2008). In our case, the essential goal is to predict biomedical relations, particularly the treatment relation. In order to implement such a system, we employ non–negative matrix factorization technique (also known as non-negative matrix approximation) which is a method in linear algebra where a matrix $\mathbf{X}^{m \times n}$ is factorized into two matrices $\mathbf{W}^{m \times k}$ and $\mathbf{H}^{k \times n}$ with a special property that all three matrices have no negative elements and $\mathbf{X}^{m \times n} \approx \hat{\mathbf{X}}^{m \times n} = \mathbf{W}^{m \times k} \times \mathbf{H}^{k \times n}$ as illustrated in Figure 2.1 where $k \ll \min(m, n)$. Particularly, nonnegativity constraint can be considered as a part-based representation in which a zero-value represents the absence while a positive value indicates the presence of some connections between entities. A more elaborate explanation is provided in Chapter 4.

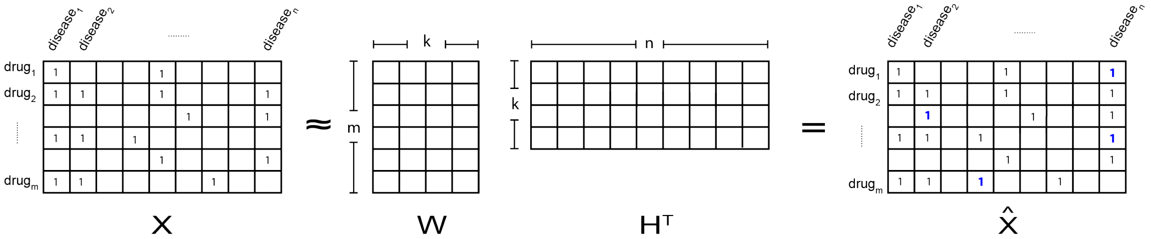

Figure 2.1: Schematic for matrix completion over non-negative matrix factorization

Matrix factorizations only have two modes or 2-way representations. Tensor factorization is considered as a higher-order extension of matrix factorization that captures the underlying latent patterns in multi-relational data sets where there is more than one predicate. Similar to matrix factorization, tensor factorization is frequently employed in several disciplines including bioinformatics, image analysis, and signal processing. There are two essential techniques for the tensor factorization: the Tucker decomposition and PARAFAC (also known as CANDECOMP/PARAFAC - or as CP) decomposition. In this dissertation, the CP tensor decomposition is used and can be thought of as singular value decomposition (SVD) analogue for tensors.

2.2.2 Neural Networks Algorithm

Neural networks (NN) algorithms are a class of supervised methods in machine learning capable of learning a subspace of nonlinear functions. A typical neural network is built with interconnected elements called artificial neurons. Neural networks can have one or multiple layers of artificial neurons. For instance, the multi-layer perceptron (MLP) is a class of feed-forward artificial neural networks and consists of at least three layers of nodes: an input layer, a hidden layer and an output layer as shown in Figure 2.2. Each layer has a number of processing units and each unit is fully connected with weighted connections to the units in the previous layer. The output layer can also include a sigmoid unit if the goal is classification (instead of regression). Since the parallel architecture in a Graphics Processing Unit (GPU) is well adapted for vector and matrix operations, NN algorithms run faster on GPU systems.

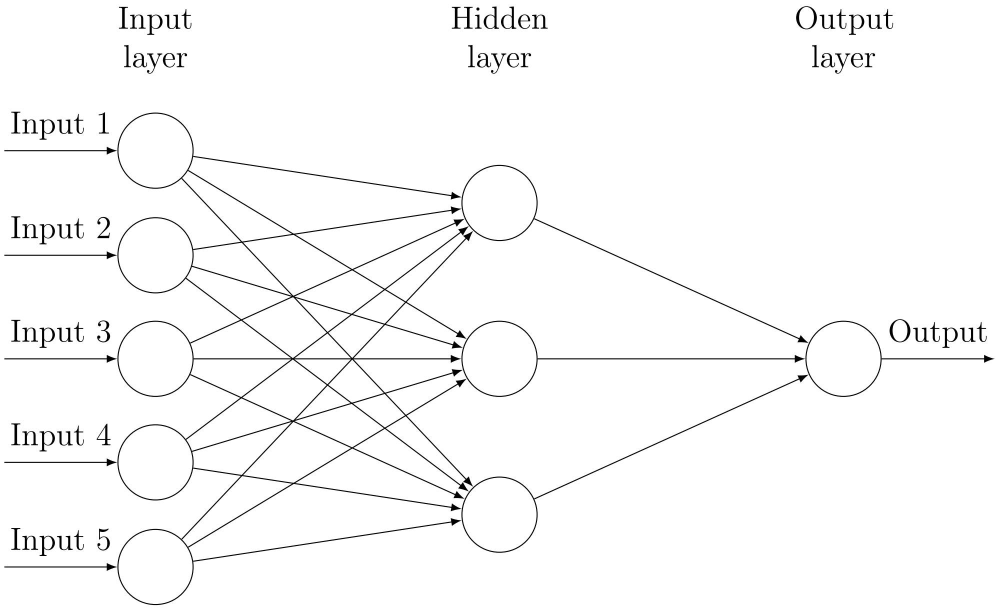

Figure 2.2: An example of a typical feedforward neural network

Neural networks models are typically built to utilize data represented as feature vectors. In Chapter 5, we handle graphs as input data, which lack input in feature vectors representation. Thus, before we examine the neural networks model that can handle graphs, it is essential to explain convolutional neural networks (CNN) first because graph convolutional neural networks (GCN) are built on the CNN principle.

2.2.3 Convolutional Neural Networks

A typical CNN model is a type of feed-forward neural network and consists of an input layer, multiple hidden layers, and an output layer. The hidden layers of the CNN basically consist of convolutional layers, an activation layer, pooling and fully connected layers. Here, the convolution layer is the core process of the CNN model which extracts features from the training dataset. For instance, if we would like to learn features for an image, then the convolution operator will collect the spatial associations between pixels with predefined convolution filters over the image (Zeiler et al., 2010). An activation layer inserts non-linearity with an activation function (*i.e.* ReLU, tanh, and sigmoid) which helps to mitigate the vanishing gradient problem which prevents updating the weights in training. The purpose of a pooling layer is to map arbitrary sized inputs to a fixed size representation and hence also to reduce the number of parameters and computational cost of the model.

The CNN architecture serves as the main inspiration for Graph Convolutional Neural Networks (GCN). The GCN architectures admit graph structures (in our case, SemMedDB knowledge graph) in a similar way in the manner that a usual CNN model takes images as input.

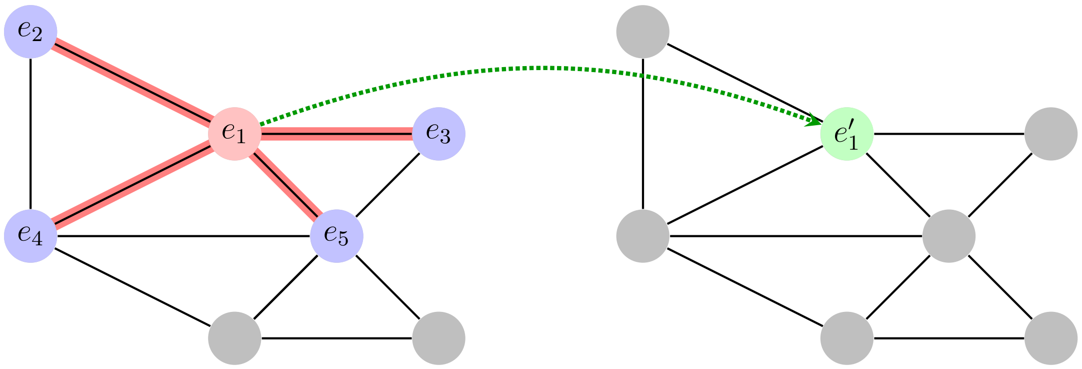

Figure 2.3: The graph convolutional networks illustration

2.2.4 Graph Convolutional Neural Networks

Traditional CNNs *convolve* over a matrix of pixels (representing the image) by fixed rectangular convolutional filters, while the GCN model uses independent convolution filters for each entity depending upon the number of neighbors of the entities in the input graph as illustrated in Figure 2.3 by Veličković et al. (2018). Technically, the convolution operation will sum all the neighboring node embeddings for both incoming and outgoing edges for every single relation that associates with the corresponding node as formulated in in the Eq. 2.1.

$$e_i^{(l+1)} = \sigma\left(\mathbf{W}_0^{(l)}\mathbf{e}_i^{(l)} + \sum_{r\in\mathcal{R}} \sum_{j\in\mathcal{N}_i^r} \frac{1}{c_i^r}\mathbf{W}_r^{(l)}\mathbf{e}_j^{(l)}\right) \tag{2.1}$$

where $\sigma(x)$ represents a non-linear activation function (can be a rectified linear unit (ReLU)), c_i^r is a normalization constant (showing the number of neighbors via the corresponding relation r), $\mathbf{e}_i^l \in \mathbb{R}^d$ indicates the latent information of node $\mathbf{v}_i$ after l iterations, $\mathcal{N}_i^r$ shows the set of neighbors with relation $r \in \mathcal{R}$ while $\mathbf{W}_r^{(l)}$ stands for the weight matrix depending upon the relation r and $\mathbf{W}_0^{(l)}$ indicates a single self-connection to each node in the graph (to retain some of its information from the previous iteration during current update).

2.3 Evaluation Metrics

The commonly used evaluation metrics in the ML applications are precision, recall and F-score. These metrics are computed scores from a confusion matrix which contains the actual true classes and the predicted classes by a classifier (Provost and Kohavi, 1998). In this dissertation, precision, recall and F-score measures are used as performance evaluation metrics. Each measure is defined as

$$P(m_i) = \frac{TP_i}{TP_i + FP_i}, \quad R(m_i) = \frac{TP_i}{TP_i + FN_i}, \quad \textit{and} \quad F(m_i) = \frac{2P(m_i)R(m_i)}{P(m_i) + R(m_i)},$$

where TP_i, FP_i, and FN_i are true positives, false positives and false negatives for the model m_i respectively. Here, precision is the ratio of correct prediction of positive test examples among all test examples predicted as positive, while recall is the fraction of predicted relevant test examples among all relevant test examples. F-score is the harmonic mean of precision and recall measures. This score takes both false positives and false negatives into account and provides a single valued estimate for the performance of a classification model.

2.4 System and Environment Details

Experiments in this dissertation were all conducted on Linux servers with the following configurations for specific efforts in the .

- Machine name: ubuntuDELVE [Chapter 3]
  - Operating system: Ubuntu 14.04.6
  - CPU(s): 64, AMD, AuthenticAMD, 2.8 GHz
  - RAM: 500 GB
  - Storage: 1.1 TB
- Machine name: kyric155 [Chapters 4 and 5]
  - Operating system: Ubuntu 16.04.6
  - CPU(s): 80, Intel, Xeon E7-4820, 2.0 GHz
  - RAM: 3 TB
  - Storage: 14 TB
- Machine name: gpu02 [Chapter 5]
  - Operating system: Ubuntu 16.04.6
  - CPU(s): 40, Intel, Xeon E5-2630, 2.2 GHz
  - RAM: 125 GB
  - Storage: 6 TB
  - GPU(s):
    - NVIDIA GM200 GTX TITAN X, 12 GB
    - NVIDIA GM200 GTX TITAN X, 12 GB
    - NVIDIA GP102 TITAN X (Pascal), 12 GB
    - NVIDIA GP102 TITAN X (Pascal), 12 GB

Chapter 3 Exploiting Semantic Patterns over Biomedical Knowledge Graphs

In this chapter, we describe a new approach for identifying plausible unknown treatment and causative relations by utilizing the graph pattern features and their extraction. Our basic intuition is simple: different entity pairs participating in a particular relation type (that is, linked via a specific predicate) are potentially connected in "similar" ways to each other where the connections are paths between them in knowledge graphs extracted from scientific literature. This is analogous to the NLP variant where a particular type of relation manifests with certain lexico-syntactic patterns surrounding the entity pair mentions in free text, the central idea exploited in distant supervision. In our approach, we need two essential components:

1. a broad scoped and large knowledge graph over which paths connecting candidate entity pairs can be obtained and

2. an approach to identify similar paths connecting entities, so we can abstract or "lift" specific paths to high level semantic graph patterns to be subsequently used as features in a supervised classifier.

3.1 SemMedDB Knowledge Graph

We build a large knowledge graph of relations obtained from SemMedDB (Kilicoglu et al., 2012; National Library of Medicine, 2016), a large database of (subject, predicate, object) relationships extracted from biomedical citations (titles & abstracts). SemMedDB is a public resource made available by the National Library of Medicine (NLM), which uses their NLP tool SemRep (National Library of Medicine, 2013; Rindflesch and Fiszman, 2003) to extract "semantic predications" from biomedical text. SemMedDB is produced by running SemRep on all biomedical citations made available thorough the PubMed search system. The relations recorded in this database are called semantic predications given SemRep normalizes textual mentions of entities to unique UMLS Metathesaurus concepts (that is, performs named entity recognition) and the predicates are also based upon those available in the UMLS semantic network (National Library of Medicine, 2003a). Each of the UMLS concepts also has at least one semantic type (National Library of Medicine, 2003b), which is essentially a classification system to categorize different biomedical entities. As such, the relations in SemMedDB represent a semantic summary of biomedical citations currently indexed by the PubMed search system. Our knowledge graph is essentially a directed graph with labeled edges formed from the relations in SemMedDB. The scope of this graph is very broad in a thematic sense given its edges are not limited to a particular biomedical topic. It is also large in that it has 14.3 million unique edges[*] connecting over 3 million nodes. It has already been used for literature based discovery and analysis of clinical documents (Cameron et al., 2013; Cameron et al., 2015a; Liu et al., 2012; Zhang et al., 2014b; Zhang et al., 2014c).

3.2 Specific Paths & Semantic Patterns

To abstract specific paths between entities over the SemMedDB graph to semantic patterns, we exploit an intuitive heuristic – simply replace the concepts along the path with their corresponding semantic type sets (given a concept can have more than one type) and retain the directions of the edges and edge labels as they are. For example, consider a sample graph showing a couple of paths between the drug `Lexapro (L)`[†] and the condition `major depressive disorder (MDD)` in Figure 3.1. We only employ simple paths (that is, without cycles) and ignore directionality when computing paths (but retaining it after paths are identified). Thus, we have the following two paths between `L` and `MDD`: (L, ingredient_of^{-1}, E, is_a, SUI, treats, MDD) and (L, ingredient_of^{-1}, E, treats, ND, treats^{-1}, SUI, treats, MDD), where the intermediate nodes are `Escitalopram` (E), `Serotonin Uptake Inhibitors` (SUI), and `Nonulcer Dyspepsia` (ND).

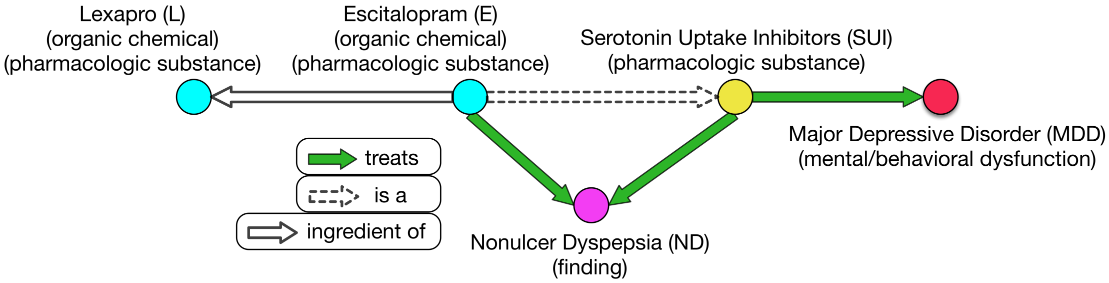
Lexapro (L)
(organic chemical)
(pharmacologic substance)
Escitalopram (E)
(organic chemical)
(pharmacologic substance)
Serotonin Uptake Inhibitors (SUI)
(pharmacologic substance)
Major Depressive Disorder (MDD)
(mental/behavioral dysfunction)
Nonulcer Dyspepsia (ND)
(finding)
treats
is a
ingredient of

Figure 3.1: A sample graph of biomedical relations

[*] Although SemMedDB (Ver. 22) has over 63 million relations, there are many duplicates given a relation can be extracted from multiple sentences due to the semantic mapping to UMLS concepts and semantic network predicates.

[†] Lexapro is the drug brand name of Escitalopram. Even though they are equivalent in pharmacology, we might say Lexapro has Escitalopram as an ingredient.

For notational convenience we encode the reverse direction with a superscript of -1 on the predicate. To obtain the patterns, we replace the specific entities with their semantic type sets. Thus, the corresponding two patterns are

$$(ingredient\_of^{-1}, \{oc, ps\}, is\_a, \{ps\}, treats) \tag{3.1}$$

and

$$(ingredient\_of^{-1}, \{oc, ps\}, treats, \{f\}, treats^{-1}, \{ps\}, treats), \tag{3.2}$$

where *oc, ps,* and f are abbreviations of the semantic types *organic chemical*, *pharmacologic substance*, and *finding* respectively. A pattern of length l (i.e., based on a path of length l) has l predicates and $l-1$ semantic types in the representation we use for this effort as shown in these examples (equations 3.1 and 3.2). Note that patterns do not include the entities being connected, but only the semantic types of the intermediate notes and the predicates along the path. By replacing specific intermediate entities with their semantic types we aim to capture high level patterns that connect candidate entity pairs. Although we just showed two paths, there are usually many others with a variety of edge types (over 50 different predicates) connecting related entities. We reiterate that our main hypothesis is that these patterns will act as highly discriminative features in identifying entity pairs that participate in a particular type of relationship. Here we clarify that although we refer to the SemMedDB graph as a knowledge graph (for general understanding), the precision of NLM's SemRep tool used to build SemMedDB is known to be around 75% (Kilicoglu et al., 2012). However, the advantage of our approach is that our prediction is not directly dependent on the correctness of each and every relation in the knowledge graph, rather on the general patterns found within it. Hence, any knowledge graph with reasonable quality will suffice although high quality graphs should yield better results. This was also observed to be the case by Cohen et al. (2014) in their distributional semantics framework.

For extraction of the paths from the knowledge graph, from our literature review, there are no efficient implementations for computing *all* simple paths of an arbitrary length between two given nodes in large graphs, although many well known algorithms (e.g., modified breadth first search) exist for identifying shortest paths. In general, finding all simple paths becomes extremely expensive with lengths greater than three simply because the number of such paths could increase drastically in dense graphs. Our implementation for lengths ≤ 3 is based on straightforward heuristics that maintain precomputed lists of neighbors for each node in the knowledge graph.

Specifically, to determine length two paths between nodes $e1$ and $e2$, we simply look at nodes in $\mathcal{N}(e1) \cap \mathcal{N}(e2)$ where $\mathcal{N}(e)$ denotes neighbors of node e. To identify length three paths, we look for edge membership for pairs in $\mathcal{N}(e1) \times \mathcal{N}(e2)$ in our knowledge graph.

In this effort, we are exclusively interested in predicting treatment/causative relationships and hence we chose this particular example for *treats* predicate from Figure 3.1. The two example patterns we show here have a nice high level meaning. In the first pattern, we see that a *pharmacologic substance* (SUI) is a hypernym for another (E) (whose main ingredient is the source (L)) and is known to treat a *dysfunction* (MDD). The second pattern is similar except that it has two pharmacologic substances (SUI and E) both treating a common second condition (ND) while one of them (SUI) treats the target condition (MDD) and the source (L) is the ingredient of the other. However, in general, the patterns themselves do not need to have interesting or meaningful interpretations, but when considered together they should be reasonably predictive of the particular predicate that is of interest to us. In this specific example, it turns out that the treatment relationship also holds for our candidate pair (L, MDD). Essentially, we expect to leverage machine learned models to automatically weight different patterns based on their predictive power rather than human experts having to manually identify interesting patterns, a highly impractical task with the explosion of biomedical knowledge.

We summarize the contributions of this chapter below:

- We propose a novel and intuitive graph pattern feature based approach to predict treatment and causative relations between any given pair of biomedical entities using logistic regression (LR) and decision tree models,

- We discuss and present details about the potential of graph patterns in terms of coverage and utility of top patterns identified through coefficients of our best LR model,

- Based on inputs from practicing physicians, we analyze false positives with high probability estimates output by our model to assess their expert based ground truth labels. We also assess the abilities of our best models to recall treatment relations from an external drug repositioning dataset.

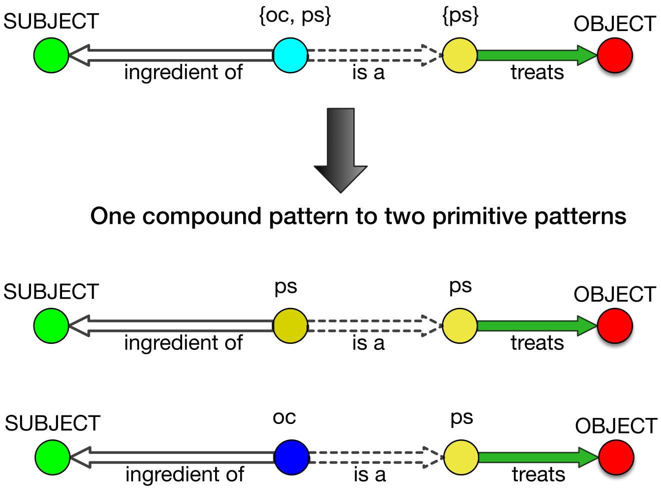

Figure 3.2: An example of primitive patterns generated from a compound pattern (from Eq (3.1))

3.3 Primitive Semantic Type Patterns

Henceforth we call the patterns discussed in Section 3.2 *compound* type patterns given an entity is replaced with the set of all semantic types assigned to it. However, there is a different way to look at semantic patterns where we split these compound patterns into potentially multiple primitive patterns to generate simpler and more generic patterns. In order to generate *primitive* patterns, we replace each set of types for the nodes in the compound pattern with just one of the constituent semantic types. Thus, we derive primitive patterns from the compound patterns simply by considering all possible combinations of constituent semantic types for each entity in the compound patterns. If we consider the first pattern in equation (3.1) as an example, the derived two primitive patterns will be as in Figure 3.2. So for a compound pattern of length l, the number of corresponding primitive patterns is $\prod_{i=1}^{l-1} |\mathcal{S}(e_i)|$, where e_i are intermediate nodes along the path and $\mathcal{S}(e)$ denotes the set of semantic types for entity e in the UMLS. Entities joined by the original compound pattern are now considered to be connected by all the primitive patterns generated from it. The primitive patterns form a more generic feature space when compared with their compound counterparts.

Complexity & Running Time Details: Finding all possible paths connecting two entities on our knowledge graph is $\mathcal{O}(t^k)$ where t is the average number of neighbors and k is the desired path length between a given pair of entities. In order to generate our features, we **have to** extract the paths for each of the training instances and convert them to semantic patterns. To test a pair of entities, we again need the extracted patterns between the entities of a given test pair. In our study, we have totally 7,000 and 2,918 pairs for *treats* and *causes* relations, respectively. For these 9,918 pairs (including training & testing), the total extraction time of the patterns was nearly two weeks.

3.4 Datasets for *treats* and *causes* Predicates

In this section we outline how we chose positive and negative examples to build the two datasets for experiments with graph pattern features introduced in Chapter 3.

3.4.1 Positive Examples from the Metathesaurus

We derive our positive examples dataset from the UMLS Metathesaurus's MRREL table (National Library of Medicine, 2009) that has over 26 million manually curated relations that are sourced from different biomedical terminologies. Among these we also have several *treats* and *causes* relations which are used for our experiments. We needed an external human vetted resource like the relations in UMLS given our knowledge graph is derived from a computationally curated relation database. We curated a set of around 7,000 unique treatment relations (entity pairs connected through the *treats* predicate) and 2,918 unique causative relations (entity pairs connected through the *causes* predicate) connecting UMLS concepts from the MRREL table. For each predicate, we divided positive example datasets into 80% (5,600 for *treats* and 2,334 for *causes*) forming the training set and 20% (1,400 for *treats* and 584 for *causes*) constituting the test set split. Although there were more positive examples in MRREL, these counts are based on pairs that had at least one path connecting them in the SemMedDB graph. This is necessary given we cannot make any prediction (given there is no information) if the entities are not connected in the graph from which we plan to extract patterns.

3.4.2 Selection of Negative Examples

Considering concerns identified in Chapter 2 to select negative examples for distant supervision we carefully choose negative examples in our dataset using the following two steps.

1. Every predicate in the UMLS semantic network, including *treats* and *causes*, has a set of domain/range semantic type constraints. That is, based on expert consultation NLM prescribes which types of entities can take the role of the

subject and object for each relation. All such possible and allowable subject-object semantic entity type combinations for each predicate are available in three tables with the SRSTR prefix (National Library of Medicine, 2009) in the UMLS. We first randomly select a pair of entities (from over 3 million unique UMLS concepts) that satisfies these domain/range constraints for the predicate for which we want to build the pattern based model.

2. For each pair selected in step 1, we check to see if the pair is connected via the predicate of interest to us either in the UMLS MRREL table or in the SemMedDB relation database. If it does not already occur in our knowledge bases, we include it as a negative example in our dataset.

This two-step process selects fairly hard-to-classify negative examples since they satisfy the domain/range constraints but do not participate in a relationship represented by the predicate for which we want to built the model. Checking for membership in both the UMLS and SemMedDB resources minimizes concerns surrounding incomplete knowledge bases. Since we want to predict treatment (causative) relations based on graph patterns, if the knowledge graph already has a *treats* (*causes*) edge between our candidate pairs, the prediction could become trivial and the whole process would be self-deceiving. Therefore, we deleted any existing *treats* (*causes*) edges between entities in all training/test positive pairs from the knowledge graph (note that negative example selection already ensures this) to guarantee a fair analysis of the predictive ability of graph patterns.

3.5 Experiments and Results

Elaborate experimentation is essential to identify performance trends across different aspects of our relation prediction problem including dataset constitution and model features and parameters. In this section we first outline some experimental configuration basics before moving on to specific models built for this effort. We start with the LR model and build upon its findings to experiment with decision tree (DT) models.

3.5.1 LR Model Configurations and Findings

We use the well known LR algorithm to predict whether an input pair of entities participates in treatment or causative relationship by building two separate binary classification models. When the number of features is much larger than the training instances, LR or support vector machine (SVM) with linear kernels are typically used.

However, SVMs do not have a straightforward probabilistic interpretation and the ad hoc means typically used to convert SVM scores to probability estimates are known to yield results that are not well calibrated (Murphy, 2012, Section 14.5). Furthermore, the coefficients of features in an LR model are estimates of log odds ratios for the corresponding features (Kleinbaum and Klein, 2010) and hence such a model lends itself to straightforward interpretation. Thus, we only use LR models in our effort.

The features for the LR models are frequencies of patterns connecting the input entity pair as discussed in the beginning of this chapter. The specific implementation used is the LR classifier based on the LIBLINEAR formulation made available through the Python scikit-learn (Pedregosa et al., 2011) machine learning library. Parameter tuning for the regularization coefficient in the LR model did not yield any noticeable gains and hence we chose to leave it at $C = 1$, the default value in scikit-learn. Performances assessment in this effort are based on standard measures of precision, recall, and F-score. All experiments were repeated using **hundred distinct 80%-20% train-test splits** of the full dataset so as to account for chance and to derive average scores and confidence intervals.

In our earlier effort (Bakal and Kavuluru, 2015), we experimented with a balanced training dataset (equal number of positive and negative instances) considering imbalanced scenarios for our test dataset. In the universe of all pairs of entities that satisfy domain/range constraints for a predicate, most are going to be false. For *treats*, an arbitrary drug-disease pair would not have a treatment relationship. So we increased the numbers of negative examples in the test to double that of the positive examples. We extended this imbalance with positive : negative ratios of 1:5 and 1:10. With a balanced training dataset, the performance gradually decreased as the test set imbalance increased. We kept the training dataset balanced to ensure that there is enough signal for the model to learn patterns for positive instances. This style of oversampling of the positive class is not uncommon in these cases where the class we care about is a rare one. In our preliminary results the performance also increased with the length of the patterns. That is, considering all patterns of length ≤ 3 resulted in better F-score when compared with considering patterns of length ≤ 2 or one.

In the experiments, we always keep the 1:10 imbalance in the test set given large imbalance is inherent to the true distribution for *treats*/*causes*. We then experiment with various imbalance scenarios in the training dataset. This is to see whether increasing the number of negative instances in the training dataset would result in performance gains on the imbalanced test dataset. The negative examples are chosen

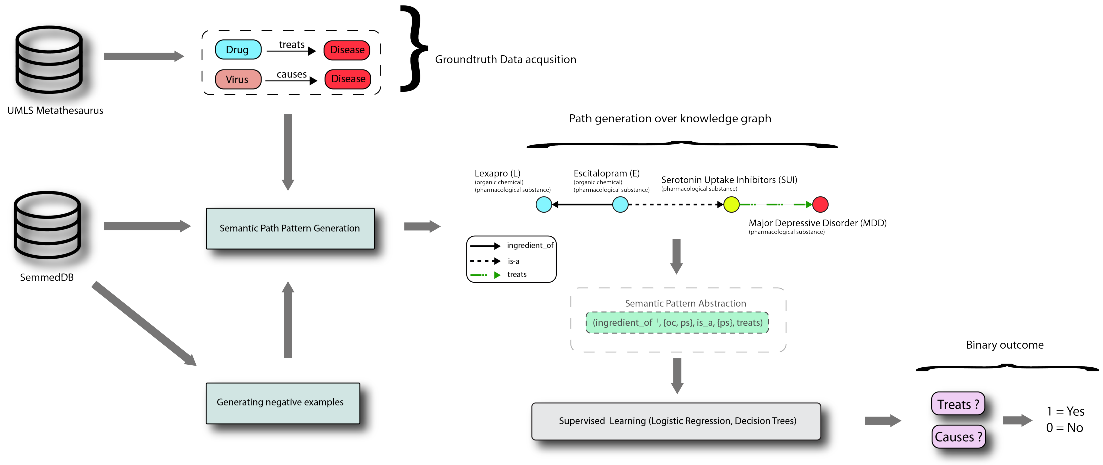
Drug
treats
Disease
Virus
causes
Disease
Groundtruth Data acqusition
UMLS Metathesaurus
SemmedDB
Semantic Path Pattern Generation
Generating negative examples
Path generation over knowledge graph
Lexapro (L)
(organic chemical)
(pharmacological substance)
Escitalopram (E)
(organic chemical)
(pharmacological substance)
Serotonin Uptake Inhibitors (SUI)
(pharmacological substance)
Major Depressive Disorder (MDD)
(pharmacological substance)
ingredient_of
is-a
treats
Semantic Pattern Abstraction
(ingredient_of $^{-1}$, {oc, ps}, is_a, {ps}, treats)
Supervised Learning (Logistic Regression, Decision Trees)
Binary outcome
Treats ?
Causes ?
1 = Yes
0 = No

Figure 3.3: Schematic of graph pattern based relation prediction

as discussed in Section 3.4.2. The full dataset size depends on the training dataset imbalance selected. For example, for the balanced training dataset and 1:10 imbalanced test set scenario, for the *treats* predicate, we have 7000 positive examples (5600 for training, 1400 for test) and 19,600 negative examples (5600 for training, 14,000 for test). When the imbalance is 1:10 in both training and test datasets, the corresponding counts are 7000 positive examples (5600 for training, 1400 for test) and 70,000 negative examples (56,000 for training, 14,000 for test). Note that these count configurations are limited by the number of positive examples available (Section 3.4.1).

Another parameter to select is the number of patterns to be included in the feature space. When classifying text with word n-gram features, researchers typically ignore all n-grams whose frequency is less than a small threshold (mostly set to five). That is, all n-grams that occurred in fewer than five documents (regardless of class membership) are ignored in populating feature vectors. We have a similar situation here with an overwhelming number of patterns of length $<= 3$ connecting entities that have a *treats* or *causes* relation. We had over 50 million unique compound patterns for *treats* and nearly 25 million such patterns for the *causes* dataset. To reduce noise and address computational efficacy concerns, we chose those patterns that occurred as connectives for at least 500 entity pairs for *treats* and 100 pairs for *causes* in the corresponding datasets. This rendered the feature spaces to manageable sizes of around 600,000 unique patterns for *treats* and 200,000 for *causes*.

The overall architecture of our method is shown in Figure 3.3. Although we are currently discussing LR models, any supervised learning algorithm can be used with graph pattern based features.

3.5.1.1 Balanced Training Dataset Scenario

As we mentioned earlier in this section, the balanced models have equal number of positive and negative examples in training dataset; the test set always has ten times as many negative examples as the positive ones to model realistic scenarios. In Table 3.1, we show the average precision, recall, and F-scores computed over hundred distinct splits of the full dataset for *treats*. The performance gains between the 1000 and 500 pattern frequency thresholds are not substantial. We see a precision gain of around 4% and a recall loss of 0.3% for each threshold when using primitive patterns over compound patterns.

Table 3.1: Balanced training data: test set scores with patterns of length ≤ 3 for treatment relations

Pattern Type	Min. Frequency: 1000			Min. Frequency: 500		
	Precision	Recall	F-score	Precision	Recall	F-score
Compound	0.675	0.926	0.781	0.683	0.928	**0.786**
Primitive	0.717	0.924	0.807	0.721	0.925	**0.810**

Performances when using primitive patterns are also superior for *causes* as shown in Table 3.2 except for the higher pattern frequency threshold of 1000. The actual F-scores are lower for causative relations when compared with treatment relations. The 95% confidence intervals we computed for F-scores have widths ≈ 0.01 when using primitive and compound patterns; thus they do not overlap for both predicates. Thus overall, primitive patterns are more effective for the balanced training dataset scenarios.

Table 3.2: Balanced training data: test set scores with patterns of length ≤ 3 for causative relations

Pattern Type	Min. Frequency: 1000			Min. Frequency: 500			Min. Frequency: 100		
	Prec.	Rec.	F-score	Prec.	Rec.	F-score	Prec.	Rec.	F-score
Compound	0.446	0.744	0.554	0.472	0.776	0.583	0.478	0.811	**0.598**
Primitive	0.400	0.736	0.518	0.510	0.756	0.609	0.546	0.791	**0.645**

3.5.1.2 Imbalanced Training Dataset Scenarios

While keeping the 1:10 positive to negative class test imbalance, we wanted to see the effect of increasing the imbalance in the training dataset in contrast with the scenario in Section 3.5.1.1. From Tables 3.3 and 3.4, we notice that the imbalance setting where $|N| = 10 \cdot |P|$ in training datasets gives the best overall F-score when compared with situations with less imbalance (including in comparison with top scores in Tables 3.1 and 3.2).

Table 3.3: Imbalanced training data: test set scores with *length* ≤ 3 compound patterns for treatment relations

Imbalance in training set	Min. Frequency: 1000			Min. Frequency: 500		
	Prec.	Rec.	F-score	Prec.	Rec.	F-score
$\|N\| = 2 \cdot \|P\|$	0.979	0.962	0.970	0.981	0.964	0.973
$\|N\| = 4 \cdot \|P\|$	0.988	0.964	0.976	0.988	0.966	0.977
$\|N\| = 10 \cdot \|P\|$	0.992	0.966	0.979	0.992	0.968	**0.980**

Furthermore, the 95% confidence interval widths for the top F-scores in Tables 3.3 and 3.4 are very small – 0.0011 (for *treats*) and 0.0036 (for *causes*). The improvements are not as substantial for *treats* but are prominent for *causes* when training set imbalance is increased; for the latter predicate, however, the recall goes down with increase in training set imbalance which is compensated by an increase in precision leading to an overall better F-score. Lowering the minimum pattern frequency yields marginal improvements for *treats* compared with corresponding gains for *causes*. Note that our improvements in Tables 3.3 and 3.4 with imbalanced training datasets are using compound patterns.

Table 3.4: Imbalanced training data: test set scores with *length* ≤ 3 compound patterns for causative relations

Imbalance in training set	Min. Frequency: 1000			Min. Frequency: 500			Min. Frequency: 100		
	Prec.	Rec.	F-score	Prec.	Rec.	F-score	Prec.	Rec.	F-score
$\|N\| = 2 \cdot \|P\|$	0.744	0.724	0.732	0.868	0.775	0.819	0.865	0.846	0.855
$\|N\| = 4 \cdot \|P\|$	0.851	0.698	0.766	0.922	0.760	0.833	0.924	0.837	0.878
$\|N\| = 10 \cdot \|P\|$	0.950	0.646	0.769	0.967	0.745	0.842	0.967	0.816	**0.885**

Contrary to our observations in balanced training dataset scenarios as we mentioned in Section 3.5.1.1, we noticed that compound patterns provided major gains over primitive patterns for the imbalanced scenarios. Furthermore, the number of patterns is substantially higher for primitive patterns (at least twice as many) leading to additional efficiency concerns. We show our observations for *causes* in Table 3.5 when using primitive patterns. When comparing these scores with those in Table 3.4, it is clear that compound patterns are better overall in imbalanced training dataset cases, which offer the best case for improving test score performances. The 95% confidence intervals we computed for F-measures in the last row and last column in Tables 3.4 and 3.5 have widths < 0.01 and hence do not overlap. Thus the improvements with compound patterns are statistically significant. Although we do not show the results here, we observed a similar trend with compound patterns outperforming primitive patterns when considering patterns of length ≤ 2 for both *treats* and *causes*. We believe this reversal in performance trend for primitive patterns is due to the fact that imbalanced training datasets lead to an explosion of unique patterns from the negative examples. When this happens, the generic and simpler primitive patterns may lose their discriminative power in comparison with the more specific compound patterns.

Table 3.5: Imbalanced training data: test set scores with *length* ≤ 3 primitive patterns for causative relations

Imbalance in training set	Min. Frequency: 1000			Min. Frequency: 500			Min. Frequency: 100		
	Prec.	Rec.	F-score	Prec.	Rec.	F-score	Prec.	Rec.	F-score
$\lvert N\rvert = 2 \cdot \lvert P\rvert$	0.531	0.685	0.597	0.626	0.710	0.665	0.667	0.728	**0.696**
$\lvert N\rvert = 4 \cdot \lvert P\rvert$	0.619	0.667	0.642	0.721	0.684	0.702	0.761	0.689	**0.723**
$\lvert N\rvert = 10 \cdot \lvert P\rvert$	0.762	0.597	0.669	0.799	0.623	0.700	0.817	0.621	**0.705**

Note that imbalanced training datasets improve precision even though we add more negative examples because the test set has always had a fixed ratio of 10:1 for negative to positive example counts. Hence a training dataset with a balanced distribution of classes will typically result in a model that will predict more instances as positive in the test set, which decreases precision. But as the training dataset imbalance is increased, the model has more negative examples to identify discriminative features and hence is trained to incur fewer FPs. This leads to increased precision.

3.5.2 Experiments with Decision Trees

In Section 3.5.1, we exclusively studied application of logistic regression models to predict relations. In this section, we discuss additional experiments we conducted with decision trees (Breiman et al., 1984) to explore nonlinear models that are also interpretable. We use the same approach as in Section 3.5.1 to come up with average scores over hundred distinct runs with 80% used for training and 20% for testing. Our results are shown in Table 3.6 for the imbalanced training dataset scenario with 1:10 imbalance in the test set (given this turned out to be the best configuration based on results from Tables 3.3 and 3.4). Given deeper trees can model more complex relationships by trading off interpretability, we experimented with scenarios where the maximum depth is restricted to five and when it is left unconstrained. As can be noticed from the table, the recall is much better when depth is not constrained. The scores are also slightly better than the best results obtained through LR models (from Tables 3.3 and 3.4).

Table 3.6: Imbalanced training data: average test set scores using decision trees with compound patterns

Predicate type	Max depth = 5			No depth constraint		
	Precision	Recall	F-score	Precision	Recall	F-score
Treats	0.998	0.815	0.897	0.994	0.987	**0.990**
Causes	0.994	0.506	0.669	0.922	0.887	**0.904**

3.5.3 Recall Analysis Using an External Dataset: repoDB

Our experiments reported thus far involved positive relations that are well known and recorded in the UMLS. Although our performance scores reported are on held-out datasets, it is possible that patterns connecting well known relations may not be present in newly discovered relations. repoDB (Brown and Patel, 2017) is a database that has over 6,000 Food and Drug Administration (FDA) approved drugs and corresponding indications collected from the regularly updated DrugCentral platform (Ursu et al., 2017). As such repoDB is expected to contain the latest FDA approved drugs. To test our best (both LR and decision tree) models,

1. We removed UMLS treatment relations from the list of all FDA approved drug-disease indications from repoDB.

2. We also removed relations whose entities are already connected through a *treats* edge in SemMedDB.

3. Out of the remaining approved drug-disease relations, we removed those that do not have at least one path connecting the involved entities in the SemMedDB graph from which we derived our patterns.

Table 3.7: Balanced training data: average recall on repoDB with graph patterns

Model	Compound	Primitive
LR	18.6%	43.1%
Decision tree	**52.9%**	50.7%

After these filtering processes, we were left with 2739 new treatment relations. We built 100 different models for the *treats* predicate based on positive examples used in Sections 3.5.1 and 3.5.2 with a fresh set of negative examples chosen for each of the models. Next, we computed the average recall by running them on these 2,739 instances we obtained from repoDB. Our results shown in Table 3.7 indicate that we are able to recall over 50% of the approved drugs that are at least connected with one path in SemMedDB. Decision trees (without max depth constraints) proved to be much better than LR models. Primitive patterns seems to help LR models while both types of patterns resulted in similar performances when using decision trees. Using imbalanced training data that gave us over 95% F-scores for held-out UMLS treatment relations turned out to be ineffective to retrieve repoDB relations. Undersampling the majority class when the minority class is of high relevance is a tested method (Wallace et al., 2011) and appears to work well for our situation too.

3.5.4 Validation of SemmedDB Knowledge Graph & Semantic Patterns

As we mentioned in Section 3.4.2, the precision of the SemMedDB knowledge graph is around 75%; therefore, we would like to demonstrate the usefulness of both the SemMedDB graph and our research idea in an alternative way of validation. In order to validate SemMedDb and show its predictive power, we pick 1,000 pairs that have at least five times direct *TREATS* and *CAUSES* relations between them for treats and causes predicates separately. Once we select the pairs, then we remove them from the original SemMedDB graph so that we will have new graphs where direct relations are not present between the pairs for each predicate. With the new knowledge graphs, we generate semantic patterns of the pairs to subject them to our 100 predictive models (with primitive semantic patterns) to obtain the accuracy even without the direct connection. The results we obtain for both treats and causes predicates are below.

Table 3.8: Validation of the SemMedDB graph with graph patterns

Predicates	Precision	Recall	F-Score
Treats	1	0.897	0.945
Causes	1	0.773	0.871

When we analyze the results as shown in Table 3.8, we can simply see that precision is 1 for each predicate because we are deliberately using only positive instances without direct relations between the pairs. Therefore, due to the absence of any FP instances, the precision stays 1. However, there are also highly predictive negative instances (selection process expressed in Section 3.4.2) in the training data. Hence, we have some FN instances as output which give rise to moderately lower recall scores. When we look at the F-Score column, the confidence intervals for treats predicate are (0.94 ± 0.002) while the confidence intervals for causes are 0.871 ± 0.005. The reason of partially lower scores for causes is that we have less number of pairs than treats predicate to encode more indicative paths & patterns. Eventually, these results demonstrate that SemMedDB is a useful knowledge source even though its precision is not perfect.

3.6 Qualitative Analyses of LR Models: Informative patterns and new hypotheses

In Section 3.5, we focused exclusively on quantitative evaluation of our methods and showed that best results are obtained by using compound patterns and imbalanced training datasets for both predicates. But it is also important to analyze the patterns qualitatively in terms of their discriminative power and their suitability in discovering previously unknown relations.

3.6.1 Exploring Highly Discriminative Patterns

In order to assess the predictive contribution of different graph pattern features, we conducted an additional experiment to identify patterns that correlate well with positive examples. During the process of building hundred different models based on hundred distinct 80%-20% train-test splits of the full datasets, we stored model coefficients for all features. Subsequently, we ranked all patterns based on the average coefficient value across the hundred models. If β is the model coefficient of a pattern in the LR model, we know e^{β} is the odds ratio of that pattern with respect to the positive class (Kleinbaum and Klein, 2010). Hence, ranking patterns in the descending order of the corresponding average model coefficient values is equivalent to ranking them based on their importance toward the positive prediction for the corresponding predicate. Thus we ranked all patterns accordingly and made them available as supplementary materials along with this manuscript. The patterns can also be searched and visualized using an online interface: http://patterns.mgokhanbakal.net/. In order to assess the sensitivity of the top patterns, we considered the top hundred patterns selected as per this ranking. For *treats*, the top 100 patterns cover 43% of the 7,000 instances. For *causes*, the top 100 connected 25% of the full positive instance dataset. This indicates that our method is able to identify high quality patterns that can be used to query knowledge graphs for generating potential new hypotheses.

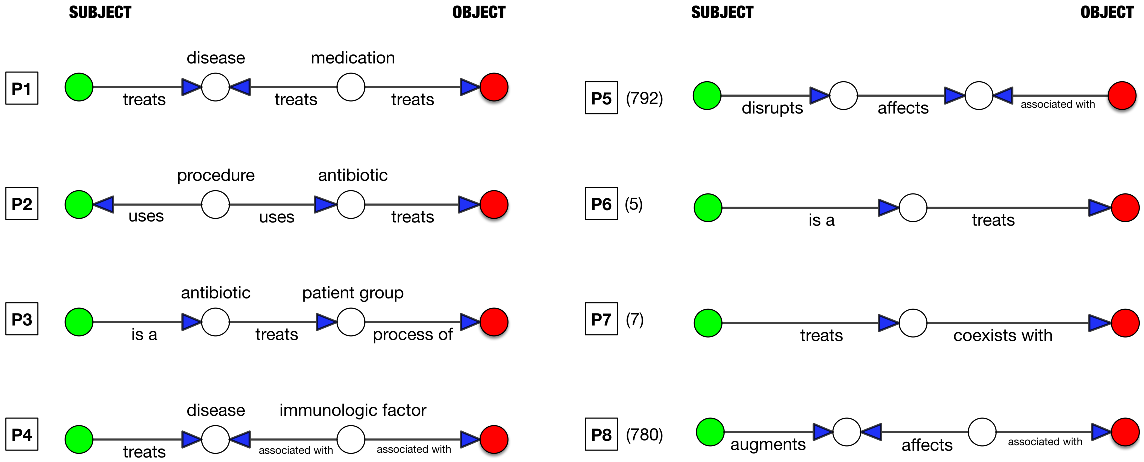

Figure 3.4: Example discriminative *treats* patterns obtained through our methods

Another objective is to manually examine these patterns and see if they are meaningful or informative in some sense. We show some interesting patterns in Figure 3.4 from our full pattern list for *treats* predicate. Patterns P1–P4 are those obtained from the top 100 patterns among nearly 600,000 unique patterns ranked. P1 indicates the situation where two drugs treat a common condition (node 2) and given one of them (node 3) treats our target condition, it is also plausible for our source substance to treat the target. P2 has a similar structure except we have a common therapeutic procedure that uses the two medications (a source and another intermediate antibiotic). P3 involves the patient group semantic type (e.g., cancer patients) that is connected to a condition via the *process-of* predicate. It also uses a class membership relation as the first edge to form a meaningful pattern connecting the instance of a class of drugs to a target condition affecting a patient group. P4 involves two conditions (an intermediate one and the target condition) that share an immunologic factor and the pattern connects the source to the target via a treatment relation involving the intermediate condition. Thus we see that patterns identified through our approach appear to have an intuitive semantic interpretation.

Patterns P5–P8 are also high scoring patterns that appeared in the top 1% of the full ranked list. We show them in the figure given a recent effort by (Cohen et al., 2014) also identified them as top scoring reasoning pathways for cancer therapies. In fact, all pathways identified by them show up in the top 1% of our ranked pattern list. We do not show the semantic types of intermediate nodes given Cohen et al. do not consider types as part of their reasoning pathways. Hence, each of their pathways can match multiple patterns in our list; hence, we show counts of our unique patterns (or unique type combinations) that match the corresponding pathway in parentheses next to the ID for P5–P8 in Figure 3.4. However, as we pointed out in Chapter 2, Cohen et al.'s work takes a retrieval approach to surface a few top patterns, while we focus on building a high accuracy predictive probabilistic model that is also interpretable through its feature coefficients.

One interesting observation here is that most of the patterns for treatment relations shown in Figure 3.4 have a *treats* edge in them. In fact, among the top 1000 treatment (causative) relation patterns 646 (241) contain a *treats* (*causes*) edge. This is not surprising for treatment relations given certain drugs and procedures treat clusters of diseases that share certain characteristics. Thus, even though the predicted relation may not be there in the SemMedDB graph, other treatment relations involving the subject medication might be indicative of its therapeutic potential for the target condition. Intuitively, this also conveys the general motivation behind computational drug repurposing efforts that are popular these days (Andronis et al., 2011; Li et al., 2016; Xu et al., 2014). Another aspect of note is that most top patterns are of length three. Of the top 10,000 patterns for each predicate, the count of length two patterns is only 24 for *treats* and 47 for *causes*. This might simply be because of the fact that, in general, paths of length three are much more common than length two associations in SemMedDB. Hence length three patterns offer a much larger feature space to exploit for our predictive models.

3.6.2 Discovering New Relations

Our evaluations thus far focused on hand curated relations already recorded in the UMLS or repoDB. However, we thought it would be more interesting to see if our approach can discover new plausible relations that are not already known. From Sections 3.5.1.2 and 3.5.2, it is clear that our approach achieves very high precision for both treatment and causative relations. However, we still have some false positives (FPs). The intuition is that high scoring FPs could actually be plausible new relations that are not already known to the medical community. To this end, we chose 10,000 new negative examples for *treats* and *causes* that are not part of the negative examples chosen to be in our training datasets used in Section 3.5 and the additional examples used in this section. We built 100 different models for each predicate changing only the negative examples as was done in Section 3.5.3. We applied these hundred models to each of the 10,000 negative examples chosen for this experiment.

Next, we needed a way to carefully choose high confidence positive predictions for expert review. For this, we finalized a list of all negative examples that were predicted as positive by at least 90 models (that is with output probability ≥ 0.5) and have an average probability estimate of at least 0.9 (overall hundred models). This process resulted in a total of 181 instances for *treats* and 138 instances for *causes*. These potentially new relations were independently reviewed by two practicing inpatient physicians (Drs. Talari and Kakani) at the University of Kentucky hospital for biomedical plausibility. After independent annotations, both physicians resolved their disagreements. 33% of *treats* FPs and 28% of the *causes* FP instances examined were deemed plausible by the physicians. Thus we are able to identify relations that are not in our knowledge bases but are still plausible. However, we needed experts to also assess novelty based on their knowledge. Although these FPs are plausible positive instances, they could just be common knowledge to experts and are simply not available in UMLS and SemMedDB. Among the *treats* instances that were deemed plausible, 68% were also identified as potentially novel findings by the physicians. This proportion is only 5.5% for *causes*; so most plausible FPs were already known to the experts despite their absence in our sources.

Among the manually reviewed FP examples, the experts chose a few plausible and novel examples for each relation (*treats* and *causes*) and came up with the corresponding plausibility explanations as follows. This was done to simulate the discovery process using our approach and offers additional evidence of practical relevance of our methods.

Plausibility of new treatment relations identified

1. **Gentamicin Sulfate** $\xrightarrow{\textbf{\textit{TREATS}}}$ **Anthrax disease**:

   Gentamycin Sulfate is an antibiotic that is used on the outside of the body (topical). It belongs to aminoglycoside class of antibiotics. It acts by disrupting the normal cycle of ribosomes, which are the structures present inside a cell. This disrupts initiation of protein synthesis inside the cells. These antibiotics are directed primarily against aerobic gram-negative bacilli class of bacteria but have limited activity against gram-positive class of bacteria. Bacillus Anthracis bacteria causing Anthrax is classified as gram-positive rod. Cutaneous Anthrax disease can be potentially treated by topical gentamicin sulfate with the above rationale.

2. **Orbifloxacin** $\xrightarrow{\textbf{\textit{TREATS}}}$ **Dysentery | Infectious Diarrhea**:

   Orbifloxacin is an antibiotic mainly used in animals. It belongs to fluoroquinolone class of antibiotics. Fluoroquinolone class of antibiotics are used in human beings for the treatment of Dysentery and Infectious diarrhea. Orbifloxacin is a fluoroquinolone antibiotic, so there is a biological plausibility for it be used in human beings for the treatment of Dysentery and Infectious diarrhea, as the mechanism of action of these group of drugs is the same.

3. **Zorubicin** $\xrightarrow{\textbf{\textit{TREATS}}}$ **Acute Myelomonocytic Leukemia**:

   Zorubicin is a medication that belongs to Anthracyclin class of drugs. Anthracyclin class of drugs are used in the treatment of cancers including leukemia. Therefore, it is biologically plausible for it to be used in the treatment of acute myelomonocytic leukemia.

4. **Ziconotide** $\xrightarrow{\textbf{\textit{TREATS}}}$ **Nonspecific Urethritis**:

   Ziconotide is a synthetic peptide related to the marine snail toxin ω-conotoxin, which selectively blocks N-type calcium channels at the cellular level. It is used in patients with chronic pain by injecting this substance into the spinal canal. With this rationale, this drug can be used to treat pain from Nonspecific urethritis as well through the same mechanism of action.

5. **Miocamycin** $\xrightarrow{\textbf{\textit{TREATS}}}$ **Staphylococcus Aureus Pneumonia**:

   Miocamycin is an antibiotic that belongs to Macrolide class antibiotics. Macrolide antibiotics have activity against many classes of bacteria including gram-positive cocci class of bacteria. Staphylococcus Aureus is a gram-positive cocci class of bacteria. With this rationale, miacomycin can be used to treat Pneumonia caused by Staphylococcus Aureus bacteria.

Plausibility of new causative relations identified

1. **Human Metapneumovirus** $\xrightarrow{\textit{CAUSES}}$ **Systemic Lupus Erythematosus (SLE)**:

   The etiology of SLE is unknown and is probably multifactorial. Interplay of genetic predisposing factors, environmental factors, and hormonal factors is thought to play a role. Among environmental factors, various viruses are thought to stimulate the body's immune network. For example, people with SLE are known to have high levels of autoantibodies to Epstein Barr virus and certain retroviruses. Thus the role of immune response to Human metapneumovirus infection in the etio-pathogenesis of SLE is a topic that warrants additional exploration.

2. **Maternal Fetal Infection Transmission** $\xrightarrow{\textit{CAUSES}}$ **Autoimmune Diseases**:

   The etiology of many autoimmune diseases is unknown. During immune development in the fetus, maturing lymphocytes in thymus and bone marrow are exposed to several antigens and those immune cells reacting to self-antigens are selectively inactivated via apoptosis (programmed cell death) or by induction of anergy. Thus the involvement of fetal infection during gestation with the process of self-antigen recognition is worth further analysis.

3. **Human Herpes Virus 6** $\xrightarrow{\textit{CAUSES}}$ **IgG Gammopathy**:

   Human herpes virus 6 (HHV-6) was first isolated in patients with lymphoproliferative disorders. HHV-6 infection has been associated with a prolonged mononucleosis like syndrome with prolonged lymphadenopathy and encephalitis. Associations between HHV-6 and diseases such as multiple sclerosis and neoplasia have been proposed but remain unproven. HHV-6 antigens and DNA have reportedly been detected in some malignant tissues such as lymphomas. Hence HHV-6 may play a role in IgG gammopathies (increased immunoglobulins belonging to Ig-G class due to abnormal proliferation of some bone marrow cells) such as Monoclonal gammopathy of uncertain significance (MGUS) deserving additional attention.

We also wanted to see if we can find examples for some of the high confidence FPs in literature (in addition to clinician assessments discussed earlier in this section). For this purpose, we used one of the high confidence FPs which is (*Ampicillin, treats, Ischemic Enteritis*). We searched the key phrase as (**Ampicillin AND "Ischemic Enteritis"**) on PubMed and found some articles mentioning the treatment connection between the entities. In this context, one of the articles found was "*A Novel Model of Ischemic Enteritis Induced in Rats by Stenosis of the Superior Mesenteric Artery*". In this article, the last sentence of the key finding section in the abstract is "The **ischemia-induced enteritis** was significantly prevented by repeated treatment with aminoguanidine (a selective iNOS inhibitor), L-NAME (a nonselective NOS inhibitor), **ampicillin**, and aztreonam (a gramnegative bacterium antibiotic), but not vancomycin (a gram-positive bacterium antibiotic)". Here, the sentence clearly conveys ampicillin has a significant role to prevent ischemia-induced enteritis. In this specific example, we can see that high confidence FPs can also be supported in medical literature.

3.7 Conclusion

Treatment and causative relations are central to knowledge discovery in biomedicine. In this chapter, we employed semantic graph patterns connecting pairs of candidate entities as the sole set of features to predict treatment and causative relations between them. We exploited a well-known biomedical relation database, SemMedDB, to build a knowledge graph with over 14 million edges extracted from scientific literature. We then used this graph to derive features and also select suitable negative training instances for predictive modeling experiments. Evidenced by the results presented in Section 3.5, we have successfully verified our hypothesis that semantic patterns over knowledge graphs can be powerful predictors of treatment and causative relations. Specifically in Section 3.5.3 we demonstrated that supervised *treats* models trained with graph pattern features can also recall newly approved drugs along with the corresponding indications from an external dataset. Moreover, we analyzed the top patterns informed by model coefficients and demonstrated their interpretability for gaining insights into the prediction process. Additionally, we sought human expert assessments to demonstrate the utility of the proposed approach in identifying potentially novel and previously unknown relations. Eventually, our results in this effort demonstrate that semantic patterns over knowledge graphs hold great promise for global relation prediction in biomedicine.

Chapter 4 Non-Negative Matrix Factorization for Drug Repositioning: Experiments with the repoDB Dataset

Computational methods for drug repositioning are gaining mainstream attention with the availability of experimental gene expression datasets and manually curated relational information in knowledge bases. When building repurposing tools, a fundamental limitation is the lack of gold standard datasets that contain realistic true negative examples of drug–disease pairs that were shown to be non-indications. To address this gap, the `repoDB` dataset was created in 2017 as a first of its kind realistic resource to benchmark drug repositioning methods — its positive examples are drawn from FDA approved indications and negatives examples are derived from failed clinical trials. In this chapter, we present the first effort for repositioning that directly tests against `repoDB` instances.

Repositioning previously approved drugs for new indications has become highly desirable in the biomedical and pharmaceutical research enterprises given expected time/cost reductions in identifying new treatment options. With recent estimates putting new drug development R&D costs over $2.5 billion per drug (DiMasi et al., 2016), repurposing has been gaining mainstream attention over the past five years. With previously approved drugs already passing the required safety tests for use in humans, the cost of repositioning is expected to be substantially lower compared with starting from a blank slate. In 2011, the National Library of Medicine (NLM) introduced `drug repositioning` (DR) as a new Medical Subject Heading (MeSH term) and as of now there are 1,161 articles tagged with it dating back to a single article from 2009. Almost 85% of these articles are published in the last five years indicating the sudden and deserved surge of interest in this area. Physicians with deep understanding of both the mechanisms of action for drugs and disease characteristics may be able to recommend off-label use (Stafford, 2008) in an *ad hoc* manner. However, this does not constitute (FDA) approved recommendation for specific new indication(s) of drugs for use in designated groups of patients. As such, DR (via FDA approval) for new indications has significant potential to impact care at a broad scale and might also result in lowered costs for patients.

Overall, we make the following contributions in this chapter:

- We demonstrate that matrix completion through NMF is a practical way for computational drug repositioning (CDR) by pruning the large space of drug candidates before further analysis.

- We evaluated our NMF models against a gold standard drug repositioning database, repoDB, and showed that the NMF values of approved and failed treatments were not overlapping in any configuration.

4.1 Related Work: Computational Drug Repositioning

Computational drug repositioning (Li et al., 2016) (CDR) is the use of informatics and high performance computing methods to prioritize candidates for new indications. With the simultaneous excitement surrounding biomedical data science and the explosive growth of publicly shared datasets, CDR methods are on the rise in the scientific community. A class of such methods exploits the notion of "similarity" between different entities involved in disease and therapeutic mechanisms. For example, shared traits among drugs including chemical structures, molecular activities, and side effects may be used to define a feature vector to represent a drug. Likewise, similarities can also be established between diseases based on established gene–disease associations or graph based proximity in disease ontologies. Zhang et al. (2014a) provide a unified framework that exploits these similarities for CDR. Another direction of CDR is exploiting available large-scale genomic data sources. For instance, Dudley et al. (2011) utilized drug-gene expression signatures to discover a potential new drug for the inflammatory bowel disease. Topological analyses of drug–target networks and target-involved pathways are another mode of identifying potential new indications (Li and Lu, 2013). Text mining programs that extract different relations from text using natural language processing (NLP) and literature based discovery approaches that build on such relations are also being employed for CDR (Andronis et al., 2011; Cohen et al., 2014). A more detailed treatise of CDR methods is available in a recent survey (Li et al., 2016).

Although CDR is gaining prominence, evaluating CDR methods can be tricky given the lack of datasets that are tailored for it. Specifically, from our literature review we were able to identify very few standardized datasets (Zhang et al., 2014a; Gottlieb et al., 2011) that are uniformly used across efforts for benchmarking purposes. Furthermore, the datasets used in prior efforts have a serious shortcoming — they only contain positive drug–disease indication pairs; and hence prior efforts assume that all other combinations are negatives, which is unreasonable and potentially rules out novel repositioning predictions as false cases. Brown and Patel (Brown and Patel, 2017) highlight this shortcoming and propose a new gold standard database called repoDB for CDR method benchmarking. repoDB draws approved indications from the DrugCentral (Ursu et al., 2017) database and failed indications from the American Association of Clinical Trials Database (the 'AACT Database' (Tasneem et al., 2012)), which is a structured version of information from NLM's ClinicalTrials.gov service. Given failed indications are part of the dataset, one can directly assess CDR methods with vetted indications and non-indications from repoDB. Since its introduction in 2017, however, we are not aware of any CDR efforts evaluating against repoDB.

In this effort, we present the first CDR attempt that directly tests against repoDB instances. First, a partially observed matrix is built using drug–disease *treatment* relations drawn from the UMLS Metathesaurus (National Library of Medicine, 2009) and those extracted using automated NLP methods and made available by the NLM as part of the SemMedDB database (National Library of Medicine, 2016; Kilicoglu et al., 2012). Next, this matrix is completed by filling unobserved cells via non-negative matrix factorization (NMF) to elicit new indications. Our method uses a small portion of repoDB as validation and uses the bulk of it for testing purposes.

4.2 Datasets

In this section, we describe the data sources from which we derive our training and testing examples. The UMLS (National Library of Medicine, 2009) and SemMedDB are our essential data resources for training instances while repoDB is our resource for the test examples. We use the terms "training" and "testing" to emphasize that this is still a (weakly) supervised method where the training instances are simply drawn from external resources both manually curated (UMLS) and automatically extracted (SemMedDB). We will briefly describe each data source in the following subsections.

4.2.1 UMLS Metathesaurus

UMLS is a longstanding terminological resource that integrates over 160 different vocabularies updated every year by the NLM. The Metathesaurus portion of UMLS aggregates equivalent concepts across multiple vocabularies and assigns to each unique concept a *concept unique identifiers* (CUI). Besides synonymous names for each concept, there are also inter-concept relations sourced from the original vocabularies. We obtained *treatment* relations from the MRREL table* in UMLS Metathesaurus

*Specifically, these are the relations where the **RELA** field in MRREL table is equal to one of these four types: *"treats", "may_ treat", "treated_ by"*, and *"may_ be_ treated_ by"* database (National Library of Medicine, 2009) version 2017AB. A total of 43,898 such relations are part of our UMLS training dataset.

4.2.2 SemMedDB – Semantic Medline Database

SemMedDB is a repository of (subject, predicate, object) triples called *semantic predications* extracted by a rule-based NLP tool SemRep (National Library of Medicine, 2013) developed by the NLM. SemMedDB is built by running SemRep over all available PubMed citations (over 27 million) where the subject/object entities are normalized to UMLS CUIs. Likewise, the predicate is mapped to a relation type from the UMLS semantic network (National Library of Medicine, 2003a). Given a predication can be extracted from multiple sentences, we also have frequency information (number of unique sentences containing it) for each SemMedDB triple. For our experiments, we curated *treatment* predications (triples where predicate = TREATS) in SemMedDB as additional training examples. As SemRep's precision is around 75% (Kilicoglu et al., 2012), we only collected predications which have been extracted at least twice, thrice, and five times to include them in the training set in various configurations (more later). Hence, we were able to obtain three different *treatment* predication sets of 55,349, 34,802 and 19,777 triples for the frequencies of 2, 3, and 5 respectively as long as **they are not occurring** in test sets from `repoDB`.

4.2.3 The `repoDB` Database

As indicated in Section 4.1, instances in `repoDB` come from DrugCentral (Ursu et al., 2017) and ClinicalTrials.gov (Tasneem et al., 2012) resources. It has a total of 6,677 approved and 3,885 failed drug–disease pairs. After removing the duplicates and the ones which appear in UMLS (given UMLS pairs will be part of the training dataset), we were left with 6,218 approved and 2,852 failed pairs. After removing pairs associated with drugs for which there is not even a single positive pair from UMLS/SemMedDB, we are left with 5,172 approved treatments (ATs) and 2,244 failed indications (FIs). This aligns with the nature of CDR to some extent — if we do not even have a single occurrence of a drug treating some disease, we may not be able to repurpose it for other conditions. This is also an inherent limitation of the matrix completion method we propose to use; if the row corresponding to a drug in the drug–disease matrix is empty, matrix completion methods cannot fill that row and hence it is impossible to come up with new indications for it (more later).

4.2.4 Generation of Randomly Selected Negative Examples

The repoDB examples are vetted ATs and FIs, identified based on clinical trials. We also wanted to build a separate dataset of random indications (RIs) which satisfy domain/range constraints for subjects/objects for *treats* predicate. The purpose is to see if our method would have a relatively easier or harder time when dealing with these when compared with FIs from repoDB. In the past we have generated such a dataset (Bakal and Kavuluru, 2015) for a slightly different task. Basically, these RI examples are created by the following steps.

- Each concept in UMLS has at least one semantic type that represents a class membership. Furthermore, every predicate in the UMLS semantic network has a set of domain/range semantic type constraints defined by the NLM based on domain expert knowledge. Based on the allowable semantic type combination for the *treats* predicate, we randomly select pairs that satisfy the domain/range constraints.
- For the set of pairs selected using the previous step, we simply remove the pairs which appear as treatment relations either in UMLS or SemMedDB. Thus, we ensure that selected pairs do not occur in our training set.

The given steps above pick fairly hard-to-predict *potentially* negative examples because they satisfy the domain/range constraints and are not present in either UMLS or SemMedDB databases. Ultimately, we obtained 3,318 examples to be used as the RI test set for the matrix completion methods.

4.3 Methods

In this section we present the NMF based matrix completion method along with our approach to configure it with different input matrices from external data sources.

4.3.1 Matrix Completion Through NMF

Matrix completion (Jannach et al., 2016) is the process of filling missing entries in a partially observed matrix. These partially observed matrices arise in many real world scenarios especially in recommender systems where preferences of people are encoded. A matrix with customers as rows and and products (e.g., movies, books) as columns is the typical setup. Given information about their prior ratings or product purchases represented as 1s in the corresponding cells, matrix completion would identify what other cells ought to be 1s — which other products would a customer likely enjoy given what they already liked. In a completely random world, there is no way to guess the new 1s. However, assuming the matrix has a much smaller rank than $\min(m, n)$ for the $m \times n$ matrix, we can use non-negative matrix factorization (NMF) (Wang and Zhang, 2013) to come up with a low-rank approximation to the original matrix with $[0, 1]$ non-zero entries in blank cells, leading to potential new recommendations. This low-rank assumption is based on the intuition that there are latent themes/traits in user preferences and a typical user's preferences are not distributed truly randomly across the product space. A similar strategy is also employed in information retrieval for latent semantic indexing (Deerwester et al., 1990) for computing document similarity through dimensionality reduction.

One can now see that the CDR problem can be modeled similarly where the training treatment relations can be used to partially fill the drug–disease matrix, with NMF filling empty cells with non-zero values pointing to potential new indications. Since this is an approximation process, the new values in empty cells will be non-zero but generally not exactly 1. Thresholding based on a validation dataset can be used to glean indications if a particular cell's value crosses the threshold. The intuition here is also to exploit potential latent themes where groups of drugs sharing certain characteristics (e.g., mechanism of action) may treat clusters of conditions with similar traits (e.g., symptoms). Given we do not know what the myriad latent themes may be, we assume a certain number of them are present — the chosen low rank — and proceed with NMF for matrix completion. Thus, given the partially observed $m \times n$ drug–disease matrix X with m drugs and n diseases, we will approximate it as

$$\underset{m\times n}{\mathrm{X}} \approx \underset{m\times k}{W} \times \underset{k\times n}{H} = \underset{m\times n}{\hat{\mathrm{X}}}, \tag{4.1}$$

where W and H are the factors with rows of W representing k-dimensional drug vectors and columns of H encoding k-dimensional disease vectors under the assumption that X has rank $k \ll \min(m, n)$. The product $\hat{X} = WH$ approximates X helping us glean new non-zero values hinting at new indications, while the rows of W and columns of H can be used to compute drug and disease similarities respectively. The objective function to find the best approximation is

$$\underset{W,H}{\arg\min} \|X - WH\| + \underbrace{\beta(\|W\|_2 + \|H\|_2)}_{\text{regularization}}, \tag{4.2}$$

where $W \in \mathbb{R}_+^{m\times k}$ and $H \in \mathbb{R}_+^{k\times n}$ and β is the weight for the regularization penalty term to handle overfitting that corresponds to large norms for W and H. Next, the

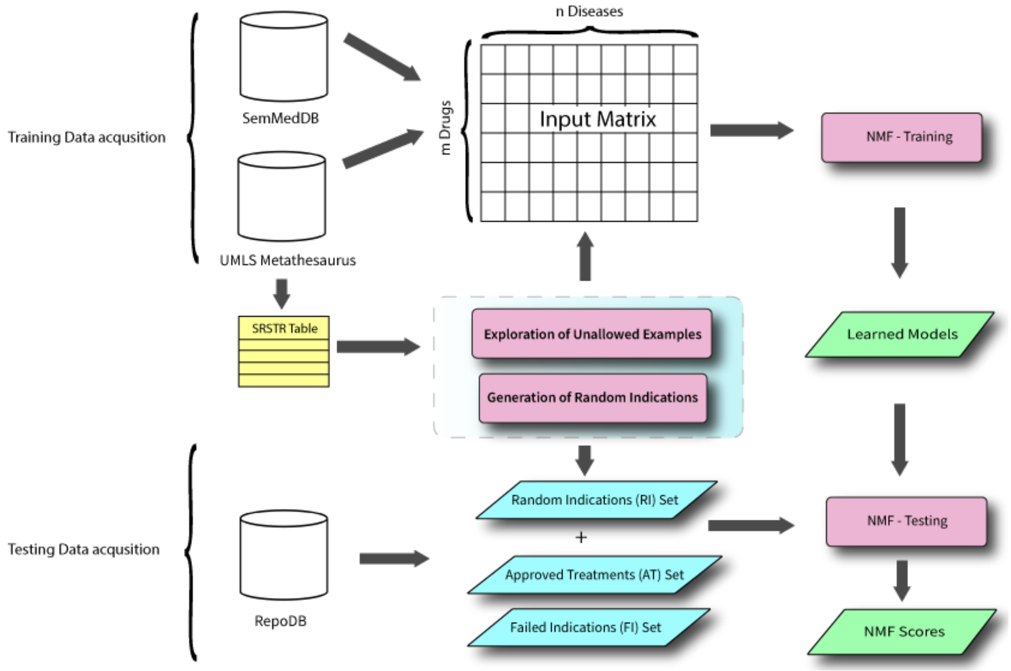

Figure 4.1: Schematic summary of the proposed NMF based CDR experiments

construction of the input drug–disease matrix X is discussed.

4.3.2 Building the Input Training Matrix

The input partially observed matrix was constructed based on treatment relations from UMLS and SemMedDB as indicated in Section 4.2. However, we needed to consider a larger matrix to fill compared with drugs and diseases of positive indications from UMLS Metathesaurus and SemMedDB. Otherwise, the test indications in `repoDB` cannot be recovered as part of the completion process. For this, we considered all allowed subject/object semantic type constraints[†] for *treats* predicate (based on SRSTR tables of the UMLS semantic network). Next, we selected all UMLS subject concepts where each has at least one semantic type that belongs to the set of allowed subject types; likewise, we aggregated all UMLS concepts where each has at least one semantic type that is from allowable object types. Based on this, 538,710 subject entities and 314,707 object entities were obtained for the input matrix. However, most subjects do not have a treatment relation with any disease as observed

[†] As an example, (*Pharmacologic Substance*, *Disease or Syndrome*) is a popular semantic type combination allowed for treatment relations. (*Antibiotic*, *Disease or Syndrome*) and (*Therapeutic or Preventive Procedure*, *Congenital Abnormality*) are less common allowed type combinations. Overall, there are 56 different allowed type combinations for treatment relations as per UMLS. There are other additional allowed types that NLM has incorporated as part of the schema design for SemMedDB and those are included for this effort as allowed combinations.

in the Metathesaurus training dataset. Hence, we removed all zero rows (hence the corresponding drugs or treatment agents) and retained only rows that are known to treat ≥ 1 disease in the columns to exploit the shared therapeutic context among the drugs. After this pruning process, we were left with 10,188 drugs (or treatment agents) and 314,707 objects[‡] to build our input matrix to be partially filled (from training datasets) and subsequently completed via NMF. Our initial motivation was using the hand-curated highly accurate treatment connections from UMLS. Since the available treatment knowledge in UMLS is limited, we add treatment connections from SemMedDB to fill the missing cells. This is also a way of **imputing knowledge** for some of the missing entries from a different source (even though the accuracy of this source is not perfect).

To populate the input matrix with training relations, we have Metathesaurus treatment relations and three different sets of semantic predications derived from SemMedDB based on extraction frequencies. We have a configuration where training relations are entered in the tables as 1s, which we call the "**binary matrix factorization (BMF)**" model. In addition to BMF, we have another configuration using SemMedDB treatment predications with their extraction frequency counts. We term this as the "**count matrix factorization (CMF)**" model to evaluate the performance when counts are used instead of Boolean indicators. Finally, as part of the input we have many unallowed input matrix cells (5,531,386 in total) that cannot be 1s because the corresponding subject–object pairs do not satisfy the domain/range semantic type constraints. For example, consider this unallowed type combination: (*Drug Delivery Device*, *Patient or Disabled Group*). Although *Drug Delivery Device* is an allowed subject type for some other type combination(s) and *Patient or Disabled Group* is an allowed object type for a different combination, this particular coupling is not allowed. However, cells corresponding to this unallowed combination exist in the input matrix given the matrix was built with all allowed subjects and objects as rows and columns, respectively. Thus, these cells corresponding to entity pairs that satisfy this type combination must be designated as unallowed. To avoid such predictions, we assign 0s to the corresponding cells of the input matrix. This is another way to further constrain the factorization process to approximate both positive and unallowed cases while estimating the unobserved cells. The unallowed examples are

[‡] Note that not all objects are diseases per se, some maybe symptoms and at times different patient groups. For example, the UMLS CUI C4316221 refers to "Patients with a diagnosis or past history of total colectomy or colorectal cancer" and can be an object of some treatment relations. To capture all latent themes we end up using all subjects and objects of treatment relations as part of the input matrix. However, for evaluation purposes, after NMF, we only look at those cells (in $\hat{X}$ from Eq. (4.1)) corresponding to approved and failed indication pairs in `repoDB`.

semantically impossible connections that need to ruled out; in this sense, they are also negative examples. We experimented with several input matrices with different numbers of unallowed cases to observe their influence on the prediction task. The levels of these unallowed examples are set to 0% (no unallowed examples), 25%, 50%, 75%, and 100% (all of them) in the input matrix. The overall framework of our approach is demonstrated in Figure 4.1.

4.3.3 NMF Experimental Configurations

We note that `repoDB` drugs/diseases for ATs and FIs are already mapped to UMLS CUIs by its creators. Hence, the matrix constructed and completed as described in Section 4.3 naturally suffices for the CDR task. When training, experiments were conducted with singular value decomposition (SVD) to identify a k value (to be used for the dimensionality in Eq. (4.1)) that minimizes the mean squared error (MSE) for the cells that are already filled in the training matrix. Since there were not any noticeable differences between MSE values with $k = 50, 100, 500$, we chose $k = 50$ for our further matrix completion experiments. Thus, results are reported for $k = 50$ for all experiments. The regularization parameter β was left at the default value (0.1) because tuning it did not yield any apparent gains. To carry out the optimization in Eq. (4.2), the open source MF library LIBMF (Chin et al., 2015) was used for incomplete matrix approximation with a total of 100 iterations. LIBMF is an efficient stochastic gradient descent based software package that runs parallel on multiple cores in a shared-memory environment.

Complexity & Running Time Details: For a given $n \times m$ matrix X, the complexity of the matrix factorization is $\mathcal{O}(kmn)$ per iteration where k is the latent dimension (Lin, 2007). This is because computing the value of a single cell requires k multiplications. The total running time (including finding the best approximation of the input matrix and the evaluation of the test set by the identified best threshold value on validation dataset) was nearly one day for each matrix completion configuration.

4.4 Results

We assess NMF results from two different perspectives. First we look the actual NMF scores produced for ATs, FIs, and RIs we created as part of Section 4.2. Subsequently,

Table 4.1: Mean of the predicted NMF scores of test sets with different configurations

Model	Training Data	Test Sets	Portion of included unallowed pairs				
			0%	25%	50%	75%	100%
BMF	UMLS only	Approved treatments	0.958	**0.268**	0.167	0.102	0.071
		Failed indications	0.980	**0.119**	0.072	0.036	0.020
		Random indications	0.995	**0.023**	0.013	0.009	0.005
BMF	UMLS + SemMedDB (MinFreq. 5)	Approved treatments	0.936	**0.564**	0.559	0.556	0.546
		Failed indications	0.957	**0.357**	0.343	0.331	0.325
		Random indications	0.987	**0.102**	0.100	0.098	0.092
BMF	UMLS + SemMedDB (MinFreq. 3)	Approved treatments	0.930	**0.614**	0.611	0.608	0.597
		Failed indications	0.954	**0.383**	0.371	0.359	0.352
		Random indications	0.983	**0.139**	0.135	0.132	0.124
BMF	UMLS + SemMedDB (MinFreq. 2)	Approved treatments	0.927	**0.650**	0.647	0.645	0.636
		Failed indications	0.951	**0.413**	0.399	0.386	0.381
		Random indications	0.976	**0.195**	0.190	0.186	0.174
CMF	UMLS + SemMedDB	Approved treatments	34.758	**7.209**	6.194	5.145	3.595
		Failed indications	30.327	1.678	3.600	0.981	0.895
		Random indications	39.603	**1.977**	0.689	0.186	0.553

we observe how these NMF scores can be used to come up with the performance metrics: precision, recall, and F-score based on ATs and FIs from `repoDB`.

4.4.1 NMF Scores for `repoDB` Examples

In Table 4.1 we show mean NMF scores for ATs and FIs in `repoDB` and the RIs we generated. The scores are real-valued numbers with which NMF fills unobserved cells as part of the training process for various configurations. Configurations differ from each other with respect to what is included in the input matrix and the proportions of unallowed pairs included as 0s. Due to the optimization in Eq. (4.2) and encoding of positive cases as 1s, the higher the value estimated for an unobserved cell, the stronger the plausibility of treatment relationship for the corresponding drug–disease pair. We make the following important observations from Table 4.1

- Only adding positive training examples to the input matrix and leaving unallowed examples as unobserved (the 0% column) leads to catastrophically bad results where the FIs and RIs are scoring higher. Hence we do not discuss any results going forward where the unallowed examples are left as unobserved. Adding additional unallowed examples (all other columns) as 0s in the input matrix shows more realistic scores following a clear pattern where the ATs score higher than FIs, which fare better than RIs. This confirms a few things: (a). The 0s inserted to account for unallowed cases are providing enough signal to guide the optimization process to distinguish between more plausible indications from random ones (which are mostly useless). (b). NMF based completion is able to score ATs better than FIs in `repoDB` where the mean AT score is 2–3 times higher than that of the FI score, demonstrating its effectiveness. (c). FIs scoring higher than RIs reflects the reality that FIs actually went through the process of clinical trials, which implies researchers felt that those pairs were plausible indications; RIs however are just random pairs of drugs and diseases. (d). Including more unallowed pairs as part of the input quickly decreases the magnitude of the scores (compare the 25% column with the 100% column); however, the relative differences between ATs, FIs, and RIs persist all across the board.

- Adding more training positives from SemMedDB (rows 4–12) increases the absolute values of the scores and also the differences in scores between ATs, FIs, and RIs, but the relative differences are the highest when using just UMLS Methasaurus relations. For example, in the 25% column, the ratio of means of AT and FIs for "UMLS only" ($0.268/0.119 = 2.25$) is higher than the corresponding ratio for "UMLS+SemMedDB (MinFreq. 5)" ($0.564/0.357{=}1.58$), which stays at a similar level even as additional relations are added (MinFreq values 3 and 2).
- Count based models (instead of the binary models) where frequencies are included in the input matrix appear not as consistent (last three rows) where for the 25% column, we notice RIs scoring higher than FIs.

We computed 95% confidence intervals that showed that the score differences are statistically significant. Here we disclose the intervals for the three rows of "UMLS + SemMedDB (MinFreq. 2)" (of the 25% unallowed cases column): **0.650** ± 0.010 (ATs), **0.413** ± 0.017 (FIs), and **0.195** ± 0.012 (RIs). In Figure 4.2, we demonstrate the confidence intervals of the corresponding case graphically. The intervals do not overlap further confirming the NMF method's functionality.

4.4.2 Performance Scores for ATs in `repoDB`

The NMF score ranges in Section 4.4.1 demonstrate that, *on average*, NMF maps ATs, FIs, and RIs to non-overlapping segments on the real number scale with high confidence. However, we still need a way to make repositioning Yes/No decisions

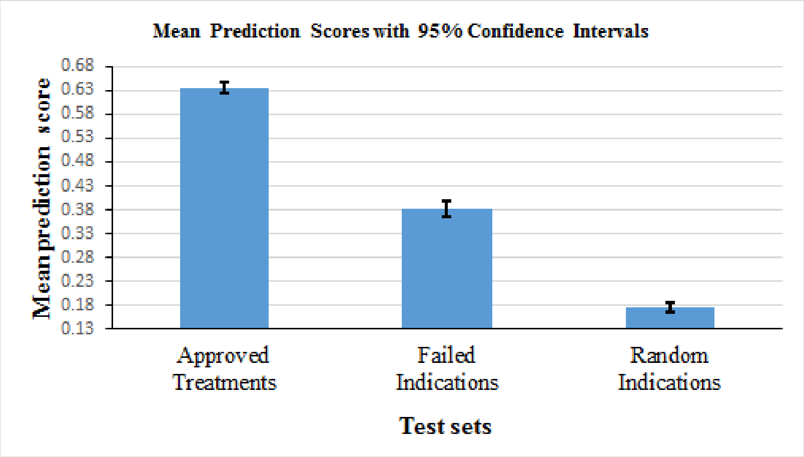

Figure 4.2: Confidence intervals for UMLS + SemmedDB (Freq.2) with all unallowed examples

at the instance level based on the score generated for a particular drug–disease pair corresponding to an entry in $\hat{X}$ in Eq. (4.1). One way to make such a decision is to choose a threshold for the NMF score and assign all pairs with scores above that threshold as new candidates for repositioning. Here we propose to do that by splitting the `repoDB` ATs and FIs into validation and test sets. We considered 20% of ATs and 20% of FIs as comprising the validation set while the rest are left for the final test[§]. We identified a threshold based on grid search over the validation dataset optimized for F-score with a small step size of 0.00001 spanning the range $[\mathcal{T}^v_{min}, \mathcal{F}^v_{max}]$ such that

$$\mathcal{T}^v_{min} = \min(\{\hat{X}_{i,j} : (i,j) \in \mathcal{T}^v\}) \quad \text{and} \quad \mathcal{F}^v_{max} = \max(\{\hat{X}_{i,j} : (i,j) \in \mathcal{F}^v\}),$$

where $\hat{X}$ is the approximation from Eq. (4.1) and $\mathcal{T}^v$ and $\mathcal{F}^v$ represent the validation datasets for ATs and FIs, respectively. This range was chosen based on the observation on the validation dataset that $\mathcal{T}^v_{min}$ is smaller than $\mathcal{F}^v_{max}$ (so there were some AT scores that were less than other FI scores). Hence choosing $\mathcal{T}^v_{min}$ as the threshold corresponds to 100% recall and selecting $\mathcal{F}^v_{max}$ leads to 100% precision. Thus by limiting the grid search to the threshold range $[\mathcal{T}^v_{min}, \mathcal{F}^v_{max}]$, we are exploring the space of compromise between perfect precision and perfect recall.

Once a threshold is chosen to make instance level decisions for test examples, it is straightforward to assess the performance of the method using traditional measures such as precision, recall, and F-score. Thus, in Table 4.2, we report the performance results for the BMF models for different configurations of the input matrix and differ-

[§] This translates to 4138 ATs and 1795 FIs in the test set and 1034 ATs and 449 FIs in the validation dataset — numbers computed based on the original `repoDB` counts from Section 4.2.3.

Table 4.2: Performance results of BMF models for approved indications in `repoDB`

		UMLS + SemMedDB(MinFreq. 5)			UMLS + SemMedDB(MinFreq. 3)			UMLS + SemMedDB(MinFreq. 2)			UMLS only			
Unallowed cases	Threshold	P	R	F-score	P	R	F-score	P	R	F-score	Threshold	P	R	F-score
25%	0.00001	0.812	0.942	0.8727	0.808	0.961	0.8787	0.800	0.964	0.8750	0.00003	0.906	0.884	**0.8952**
50%	0.00001	0.811	0.942	0.8723	0.808	0.961	0.8787	0.800	0.964	0.8750	0.00004	0.906	0.883	0.8950
75%	0.00001	0.812	0.941	0.8721	0.808	0.961	0.8787	0.800	0.964	0.8750	0.00002	0.905	0.878	0.8916
100%	0	0.723	0.981	0.8328	0.714	0.996	0.8322	0.709	0.996	0.8287	0	0.840	0.909	0.8736

ent levels of included unallowed examples. The first observation is that the thresholds selected are all very close to zero indicating that boundary case NMF scores were close to zero for ATs across all configurations. The thresholds are identical for configurations with SemMedDB examples but change slightly for UMLS-Only case. We notice that the best F-score of 0.895 (first row, last column) is obtained for UMLS-Only input matrix with 25% unallowed example constraints. This may be explained from the biggest relative difference between mean AT and FI scores for this configuration from Section 4.4.1. However, adding SemMedDB training instances (with minimum frequencies 2 and 3) seems to lead to a potentially more desirable compromise with recall around 96% and precision over 80%. The results for count based CMF models were disappointing as shown in Table 4.3. There is no clear pattern as to how the scores are spread with regards to different configurations and overall performance is all across the board inferior when compared to BMF models especially with substantially lower precision values.

Table 4.3: Performance results for CMF models over `repoDB`

Unallowed cases	Threshold	P	R	F-score
25%	0.12162	0.714	0.957	**0.8185**
50%	0.08999	0.698	0.968	0.8119
75%	0.00009	0.692	0.973	0.8093
100%	0.005	0.703	0.963	0.8131

4.4.3 Prediction Role of Chronological Treatment Knowledge for Drugs

As we mentioned before, the repurposing of a drug is a convenient way of discovering new treatment options for diseases. In this context, many drugs have been effectively applied to treat more than one condition. Hence, we conduct additional experiments to measure the influence of prior treatment knowledge for the drugs in ATs. When we check the AT examples, we notice that there are 394 unique drugs treating more than one condition (out of 634 pairs).

Since it is hard to trace years of approval for the same drug for different indications[¶], we aggregate the list of publication years of PubMed articles containing the corresponding TP pairs. Since treatment pairs can appear in multiple publications in different years, we collect the drug–disease AT pairs in $\mathcal{P}_{AT}$ where the corresponding publications for the disease are chronologically later than PubMed articles for other diseases the drug treats. As a formal representation,

$$\mathcal{P}_{AT} = \{(d, c) : (d, c) \in AT,\ minDate(d, c) > maxDate(d, c'),\ \forall_{c' \neq c}\,(d, c') \in AT\}, \quad (4.3)$$

where AT is the set of ATs and $maxDate(d, c)$ and $minDate(d, c)$ are functions that indicate the latest and earliest publication dates, respectively, where the corresponding AT pair (d, c) was found to be in a treatment relation as per SemMedDB. It is clear to see there is at most one pair $(d, c) \in \mathcal{P}_{AT}$ for any drug d. $\mathcal{P}_{AT}$ is our final test set that contains the latest discovered TP pairs where there can be only one test pair for each drug as formulated in the Eq. 4.3. Eventually, we obtained 147 pairs in the final test set $\mathcal{P}_{AT}$ when we examine the list of drug-disease pairs with their extraction years. After identifying the temporal test sets, we included the rest of the AT pairs as the prior knowledge for the temporal test set. Hence, we had 487 additional training treatment pairs.

Table 4.4: Mean of prediction scores of 147 temporal test pairs with and without temporal data

	BMF Models						CMF Models	
	UMLS + SemMedDB(Freq.5)		UMLS + SemMedDB(Freq.3)		UMLS + SemMedDB(Freq.2)		SemMedDB extraction frequency count	
Included Exceptions	w/o Temp.	w/ Temp.	w/o Temp.	w/ Temp.	w/o Temp.	w/ Temp.	w/o Temp.	w/ Temp.
25%	0.7559	0.7922	0.8495	0.8568	0.8745	**0.8756**	1.718	2.602
50%	0.7518	0.7877	0.8458	0.8534	**0.8726**	0.8724	1.588	2.207
75%	0.7480	0.7840	0.8430	0.8493	0.8703	0.8709	0.872	2.856
100%	0.7378	0.7728	0.8261	0.8366	0.8587	0.8625	**0.990**	0.796

In Table 4.4, we demonstrate the average prediction scores of the temporal test set for the BMF and CMF models. In the BMF models, we can clearly see that using prior knowledge data yields higher scores except for the case using minimum

[¶] Confirmed by Dr. Oleg Ursu (University of New Mexico), an admin for DrugCentral. He also informed that a more elaborate effort is needed to glean years for new indications by querying the `Drugs@FDA` database to identify Type 6 or Type 9 approvals which designate new indications.

frequency 2 with 50% of unallowed examples. In addition, increasing the number of unallowed examples decreases the scores regardless of prior temporal knowledge. In the CMF scenario, we also have higher scores except for the case using all the unallowed examples when we exploit the prior knowledge of the test pairs. Consequently, we noticed that the utilization of previously known treatment examples in most of the models improved the prediction scores of the AT test pairs.

4.5 Discussion

As CDR efforts continue to rise, it is critical to have benchmarking datasets that are realistic in terms of representation of both approved indications and failed indications. `repoDB` is first of its kind dataset that creates such an opportunity to conduct comparative evaluations of CDR methods on a publicly available gold standard dataset.

4.5.1 Main Takeaways

Matrix completion through NMF based low-rank approximation is an effective method for CDR based solely on datasets of previously approved drugs and corresponding indications. Actually, in this manuscript, we only use public data sources of treatment relations in the form of hand curated UMLS Metathesaurus relations and those extracted with NLP from PubMed citations (from SemMedDB). As such, these are imperfect resources (especially SemMedDB) and may not necessarily constitute FDA approved drugs. Results still show that among recoverable ATs from `repoDB`, we achieved F-scores close to 90% with the highest F-score achieved with just UMLS relations as input. Using both SemMedDB and UMLS relations helps achieve a better compromise between precision and recall with over 96% recall at 80% precision. The mean NMF score for FIs is at least twice as large as that for RIs, indicating that FIs are indeed much tougher to distinguish from ATs compared with randomly generated pairs. A critical enabler was the encoding of unallowed pairs (derived with incompatible semantic type constraints from UMLS semantic network) as zeros imposing additional structural constraints on the input matrix to be approximated. However, imposing constraints from *all* unallowed pairs could be detrimental by leading to a 3% recall gain with a 10% precision drop. Experiments showed that introducing 25% of the zeroes from unallowed pairs leads to better outcomes and is computationally less expensive. Count based models that consider frequency from SemMedDB substantially underperform compared with simpler binary models. Overall, NMF based methods applied to carefully curated external knowledge sources constitute a practical approach towards CDR.

Next we discuss some examples of correct predictions made by our approach. In our training dataset we see the drug vincristine treating *malignant neoplasms, follicular lymphoma*, and *Hodgkin disease* and another drug doxorubicin treating the general condition of *malignant neoplasms.* After matrix completion, we saw high values of 0.89 and 0.93 for the entries (doxorubicin, *follicular lymphoma*) and (doxorubicin, *Hodgkin disease*) respectively, which are approved indications in `repoDB` that were never encountered in training data and were blank cells before the training process. Similar new correct predictions are also made for (bleomycin, *follicular lymphoma*) and (bleomycin, *Hodgkin disease*). Next, although, the count based CMF method underperformed overall, there were cases where it lead to correct predications when the binary approach did not. For example, (betamethasone, *berylliosis*) and (bleomycin, *malignant head and neck neoplasm*) are approved indications that were missed by the BMF approach but are recovered by the CMF method. Thus there may be some complementary traits in how the BMF and CMF approaches predict that need further examination toward building an ensemble method.

We set out to explore reasons for errors — false positives (FPs) and false negatives (FNs) — incurred by the NMF models in the context of information available about the corresponding drug–disease pairs. To this end, we examined connectedness of FP and FN pairs in the SemMedDB graph, which essentially conveys the potential shared context between associated entities. In our prior work (Bakal et al., 2018), we identified graph patterns over the SemMedDB graph that are highly indicative of treatment relations using model coefficients of a logistic regression (LR) model[‖]. For FPs of the NMF model, we noticed that there were tens of thousands of highly predictive short paths (length ≤ 3) connecting the corresponding drug and disease CUIs indicating that there are many shared neighbors; some of this neighborhood information is encoded in the input matrix, which could have led to positive predictions. This is also not surprising given FPs are essentially failed cases in `repoDB` but were deemed plausible enough for researchers to launch clinical trials. For FNs, we found relatively fewer and sometimes no such predictive paths in SemMedDB connecting associated entities. For example, for the approved `repoDB` indication (Dexamethasone, Branch retinal vein occlusion with macular edema), the drug and disease were not connected

[‖] The LR model, while being effective, is computational prohibitive at times given the explosion of numbers of paths connecting entities in a large graph such as that built from SemMedDB. Our foray into NMF is motivated by these efficiency constraints of the graph pattern based approach.

in the SemMedDB graph using LR model's top predictive patterns. Without much shared context, NMF appears to struggle to elicit positive indications for such pairs. We plan to pursue a more detailed error analysis involving physician experts, which may yield additional insights on potential reasons for errors.

4.5.2 Limitations and Future Directions

This current effort is not without a few limitations, which also point to interesting future research directions for CDR experiments with `repoDB`.

- The method in this effort is clearly not a silver bullet for CDR. `repoDB` does enable excellent benchmarking but in general scientists are often looking at a particular disease that they want to treat. Hence, for disease specific CDR, more sophisticated methods involving gene expression datasets and methods that consider integration of various modalities of information specific to the disease may be needed, as indicated in other prior efforts (e.g., Nagaraj et al. (Nagaraj et al., 2018) for cancer). However, our method can be an effective initial step in pruning the space of candidates before more sophisticated methods that require more complex modeling and disease specific information can be applied.

- As discussed earlier, the count based CMF models' performance was underwhelming (Table 4.3) when compared with the binary models even though the counts capture additional information about prior knowledge being incorporated into the input matrix. One reason for this could be that we simply employed raw frequencies of treatment predications in SemMedDB instead of standardizing counts using well-known methods (Berg et al., 2006) (e.g., mean centering, min-max scaling, log transformation). Using raw frequencies may have lead to potential ill-conditioning that needs to be countered with appropriate pre-processing and/or using more sophisticated methods (Cichocki and Zdunek, 2006). These experiments will be part of future extensions of our work.

- Matrix completion methods cannot fill a row that does not have at least one nonzero entry. In our case, this means, a drug for which we do not have at least one known treatment relation cannot be linked to new indications with NMF. However, this can be remedied by moving from matrices to tensors with additional relations between entities connected with other predicates including *prevents*, *diagnoses*, *affects*, and *causes*. Using tensor factorization (Luo et al., 2016), even for a drug with no existing treatment relations, using multi-hop indirect connections, it is possible to elicit a new indication. Similarly, with recent deep learning advances, embedding nodes and edges of the larger SemMedDB graph (including edges arising from other predicates besides *treats*) with graph neural networks can offer a different way for knowledge base completion (Schlichtkrull et al., 2018). We will discuss those approaches in the next chapter.

4.5.3 Benchmarking

To enable future comparisons with our results, we provide the validation/test set splits of `repoDB` drug–disease pairs used in this study: `https://github.com/bionlproc/nmf-repoDB-benchmarking`. This will be important for direct comparisons by other researchers using the `repoDB` dataset, especially given we had to resort to using a subset of `repoDB` (owing to issues with lack of training instances for certain drugs without a single human vetted treatment relation).

4.6 Conclusion

With valuable time and cost savings in the offing, CDR efforts are expected to increase in the future. With lack of datasets modeling both positive and failed indications, it is encouraging to notice that datasets such as `repoDB` are being created. However, it is also important to start comparing methods against such datasets for robust assessments of different methods. In this chapter, matrix completion through NMF was used to directly predict `repoDB` approved indications by using publicly available treatment relations. F-scores close to 90% were obtained with various training configurations with this method showing its strong potential for practical applications. Validation and test splits of `repoDB` used as part of this effort are made available to facilitate direct comparisons with our results by other researchers in the CDR community. More sophisticated methods such as tensor factorizations and neural graph embeddings may hold the promise of recovering novel indications for drug compounds that have not yet been approved for any known conditions. We believe this is the first attempt to employ `repoDB` for CDR purposes and hope that this will trigger more attempts to pursue this line of work toward rigorous benchmarking.

Chapter 5 Multi Relational Data Analysis for Drug Repositioning

5.1 Motivation

As we discussed in Chapter 4, the matrix completion approach does not utilize the indirect connections over multiple biomedical relation types between drug and disease candidate pairs. To address this weakness, we utilize the available treatment predications in UMLS and other essential biomedical relation types including *prevents*, *diagnoses*, *affects*, and *interacts_with* extracted from all of PubMed citations in SemMedDB knowledge base. For this purpose, in this chapter we conduct experiments exploiting the tensor factorization (TF) algorithm with the 3-way data tensor input and the graph convolutional neural network (GCN) approach embedding entities and relations in a vector space.

The TF approach has been successfully applied to diagnostics of test results, analysis of genomic data, and completion of the electronic health records in the biomedical domain (Luo et al., 2016; Roy et al., 2014; Wang et al., 2015). Here, the advantage of the TF method is that the learned low dimensional latent embeddings capture the underlying patterns among the tensor entries for relation prediction and other various tasks such as document clustering and recommendation systems. With this motivation we employed a TF algorithm on the input tensor built with the biomedical entities connected with multiple biomedical relations. The relations used in both methods are listed in Appendix A.

Vector embeddings of the nodes and edges in a knowledge graph (a.k.a. - knowledge graph embedding) is extensively studied for the knowledge graph completion task by several efforts (Nickel et al., 2011; Yang et al., 2014; Wang et al., 2014; Ji et al., 2015; Trouillon et al., 2016). Similarly, the GCN technique is also utilized for several tasks such as discovering polypharmacy side effects for drugs, learning molecular fingerprints for drug efficacy, node classification, and predicting relations with multi-relational data (Zitnik et al., 2018; Duvenaud et al., 2015; Kipf and Welling, 2017; Schlichtkrull et al., 2018). Here, we utilize the embeddings of the semantic predications from SemMedDB and UMLS repositories. On top of that, we exploit the GCN algorithm which enhances the entity embedding characteristics by considering the neighborhood information. The ultimate goal of this chapter is to predict approved treatments (ATs) and failed indications (FIs) in `repoDB` with tensor factorization and GCN approaches. Moreover, we demonstrate that these approaches are efficient on the relation prediction task with satisfactory performance results.

Overall, we make the following contributions in this chapter:

- We conduct experiments with TF and GCN algorithms to fully exploit the SemMedDB knowledge base by considering indirect connections over the fundamental biomedical relation types as listed in Appendix A.
- We evaluate the models and compare results with the NMF model on the `repoDB` dataset. Besides, we present the advantages of the TF and GCN models in recovering new relations missed by the NMF approach.
- We obtain the best performing model with an ensemble of NMF, TF, and GCN models for the `repoDB` pairs. In addition, we conduct error analysis with the FP and FN predictions of the ensemble model by collaborating with two physicians.

5.2 Datasets

In this section, we describe the data sources for our training and testing examples to evaluate models.

5.2.1 Training Set

SemMedDB and UMLS Metathesaurus are our essential data resources for the training examples while `repoDB` is our main resource for the test examples. In order to obtain the most important predications, we filter them based on the predicates appearing in the highly indicative semantic patterns* of treatment relations in (Bakal et al., 2018) and the set of predicates of interest in the effort by Cohen et al. (2014). Essentially, we identified 20 major relations as listed in Appendix A. In our experiments we collected a total of 9,860,059 biomedical predications to use in the training set.

5.2.2 Test Set – `repoDB`

As we mentioned in Chapter 4, we totally have 6,218 approved treatments (ATs) and 2,852 failed indications (FIs) pairs after removing the duplicates and the pairs which appear in UMLS metathesaurus. Unlike the NMF experiments reported in the previous chapter, we retained all available test pairs because each entity of the pairs participates as either subject or object in the training pairs.

*The predicates were selected from the top 100 patterns that were ranked by the logistic regression coefficients.

5.3 Methodologies

As we mentioned in Section 5.1, we conducted experiments using TF and GCN algorithms to predict approved and failed pairs in `repoDB`. To that end, next we will briefly explain the technical details of both approaches.

5.3.1 Tensor Factorization Technique

Multi-relational data can be embedded into a multi-dimensional representation such as a tensor. In this sense, we can generate a three-dimensional input tensor structure with the biomedical predications to run the TF algorithm. From that angle, we can simply look at Figure 5.1 to see how we build the mentioned tensor structure with the biomedical predications. When we look at the tensor figure, we can see that the structure is a three-dimensional (symbolized as $\mathcal{X}^{\,I \times J \times K}$) representation in which we embed the predications with multiple relations. The y-dimension presents the subject entities while the x-dimension stands for the object entities, and the z-dimension indicates different relation types such as *treats*, *causes*, and *prevents*. Briefly, TF can be considered as a higher-order extension of matrix factorization that captures the underlying latent patterns in a multi-relational dataset. To this end, each cell in the tensor presents how many times a relation occurred between the corresponding entities via the particular relation in SemMedDB knowledge base. Nevertheless, many of the cells stay blank because those predications do not exist or are unknown. Therefore, our goal is to discover whether there is a possible relation for the blank cells.

We exploit the **non-negative tensor factorization** (NTF) technique. Here, the non-negativity constraint plays a major role in analyzing non-negative data for the many practical problems including image processing and text data mining. In our case, the TF method allows us to detect new context-dependent discoveries of biomedical relations among entities. The motivation here is that the underlying data stack is non-negative. Hence, we employ the TF algorithm as the interpretation of the prediction results is more practical than the usual TF scenario.

To identify the best approximation of a given tensor, there are two well-known techniques as well as their close variations: Tucker decomposition and CANDECOM/PARAFAC (CANonical DECOMposition/PARAlel FACtors – CP) decomposition (Kolda and Bader, 2009). In this research, we utilize the CP decomposition as it provides a more feasible factorization. Here, the CP tensor decomposition can be thought of as singular value decomposition (SVD) analog for matrices. SVD decom-

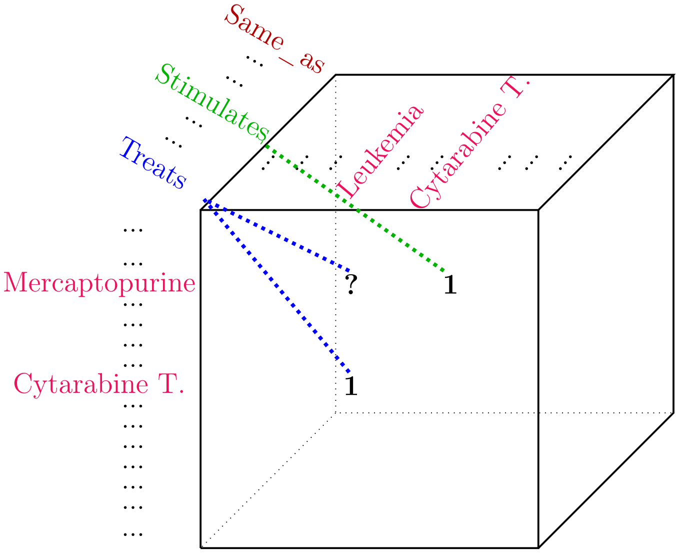

Figure 5.1: Third order tensor structure representation ($\mathcal{X}^{\ I \times J \times K}$) for embedding biomedical predications.

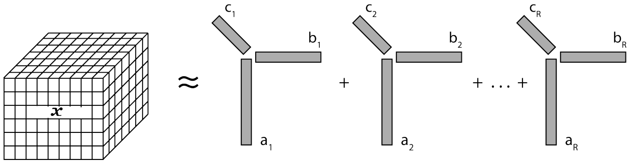

Figure 5.2: A graphical representation of CP-decomposition mechanism for a three-way tensor as a sum of rank-one tensors. (Acar et al., 2011)

poses a matrix as a sum of rank-one matrices while the CP technique decomposes a tensor as a sum of rank-one tensors (Cichocki et al., 2009) as represented schematically in Figure 5.2.

The example tensor structure $\mathcal{X}^{I \times J \times K}$ is decomposed as three factor matrices $\mathbf{A}^{I \times R}$, $\mathbf{B}^{J \times R}$, and $\mathbf{C}^{K \times R}$ such that

$$x_{i,j,k} \approx \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr} \quad \text{for} \quad i = 1, \dots I, \ \ j = 1 \dots, J, \ \ k = 1, \dots, K, \tag{5.1}$$

where R denotes the size of latent dimension while I, J, and K are the numbers of rows of the factor matrices. For the best approximation, the optimization problem in Eq. 5.2 is solved with the CP decomposition.

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \|\mathcal{X} - \underbrace{[\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]}_{\substack{\text{Approximated}\\ \text{tensor}\\ \hat{\mathcal{X}}}} \|_F^2 \quad : \mathbf{A} \in \mathbb{R}_+^{I\times R}\ , \mathbf{B} \in \mathbb{R}_+^{J\times R} \text{ and } \mathbf{C} \in \mathbb{R}_+^{K\times R} \tag{5.2}$$

In Equation- 5.3, for factor matrices, the parameter R denotes the number of components while I, J, K are the numbers of rows in the factor matrices and correspond to subject entities, object entities, and relations respectively.

$$f(\mathcal{X}, \mathbf{A}, \mathbf{B}, \mathbf{C}) = \frac{1}{2}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\left(x_{ijk} - \sum_{r=1}^{R} a_{ir}b_{jr}c_{kr}\right)^2 \tag{5.3}$$

Given sparse tensor data, we need to take care of over-fitting issue for better generalization. As we mentioned in Section 4.3.1, we have a regularization part that slightly modifies our optimization problem (Acar et al., 2011). Furthermore, in order to compute the nonnegative component matrices A, B, C, we will apply constrained optimization approach by minimizing a suitably designed cost function. Typically, we minimize the following cost function

$$\min_{A,B,C} \|\mathcal{X} - [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]\|_F^2 \quad + \underbrace{\frac{1}{2}(\lambda_A\|\mathbf{A}\|_F^2 + \lambda_B\|\mathbf{B}\|_F^2 + \lambda_C\|\mathbf{C}\|_F^2),}_{\text{Regularization part}} \tag{5.4}$$

where $\lambda_A, \lambda_B, \lambda_C$ are nonnegative regularization parameters. The most popular method to optimize is to apply the alternating least squares (ALS) technique. In this method we compute the gradient of the cost function with respect to each individual component (factor) matrix (supposing that the others are fixed and independent). In this sense, among the updating steps, ALS algorithm guarantees minimization of the cost function, until convergence. The main advantage of ALS algorithms is high convergence speed and its scalability for large-scale problems.

The main purpose is to identify the best approximation of the input tensor to analyze the prediction values of the test pairs. All the blank cells will be filled either with a non-negative real value or zero once we identify the best approximation similar to the NMF approach. Hence, the prediction task of the missing relations can be expressed as filling the blank cells in the input tensor such that the predicted values would be consistent with the existing training examples. Here, we use an open source TF library, Splatt (Smith et al., 2015) for the approximation of large

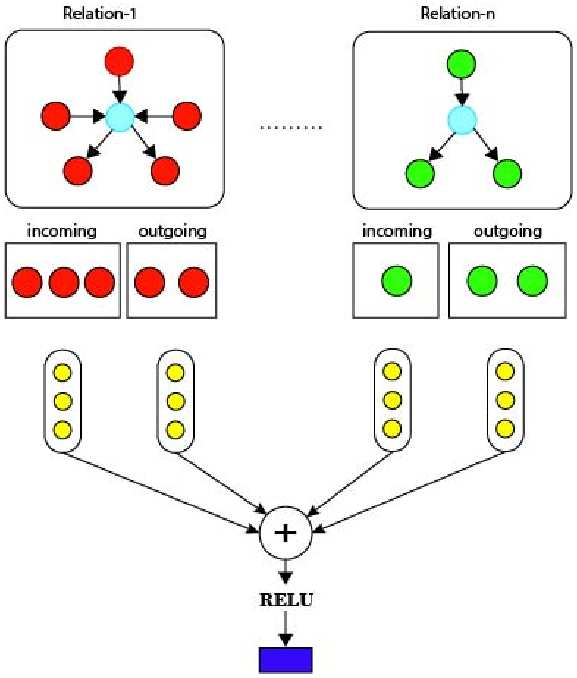

Figure 5.3: A simple graph convolution for a node for graph embedding

incomplete tensors. Technically, the Splatt library takes the advantages of shared and distributed memory parallelism concepts.

5.3.2 Graph Convolutional Neural Networks Technique

Knowledge base completion or in other words prediction of missing links in knowledge bases is one of the recently popular challenges in data science (Bordes et al., 2013; Socher et al., 2013). In this sense, Bordes et al. are the pioneers of the link prediction task on knowledge graphs (such as Freebase, WordNet, DBpedia and etc.) to predict new predications (facts) by exploiting deep learning techniques (Goodfellow et al., 2016). Technically, they embed both entities and relations into the same vector space. In this context, entities and relations are embedded as vectors such that $\mathbf{e_i}, \mathbf{r_j} \in \mathbb{R}^d$ (where d is the embedding size). The central idea behind the work is that relations are called as transitions from an entity to one another so that s + r ≈ o where (s, r, o) is a triplet (predication). Even though the idea itself is an elegant solution, it does not involve the neighborhood association data for the entities. Hence, utilizing the graph convolutional neural networks (GCN) technique on SemMedDB knowledge graph with the embeddings of the entities and relations is the primary inspiration for our GCN based experiments.

Technically, convolution operation sums all the neighboring node embeddings for both incoming and outgoing edges for every single relation that associates with a node. Unlike using solely randomly generated embeddings for the entities, we exploit the convolution operation for a node with its neighbors as depicted in Figure 5.3. In our case, we embed the entities and relations of the predications in SemMedDB and UMLS knowledge bases. Each convolution will be occurring in a neural network layer and the collected information will be passed through the links connecting the nodes in the graph. Here, in the first layer the entity embeddings are updated via

$$e_i^{(l+1)} = \sigma\left(\mathbf{W}_0^{(l)}\mathbf{e}_i^{(l)} + \sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{c_i^r}\mathbf{W}_r^{(l)}\mathbf{e}_j^{(l)}\right) \tag{5.5}$$

where $\sigma(x)$ represents a non-linear activation function – a rectified linear unit (ReLU), c_i^r is a normalization constant (showing the number of neighbors via the corresponding relation r), $\mathbf{e}_i^l \in \mathbb{R}^d$ indicates the latent information of node $\mathbf{v}_i$ with the dimension d at the layer l, $\mathcal{N}_i^r$ shows the set of neighbors with relation $r \in \mathcal{R}$, while $\mathbf{W}_r^{(l)}$ stands for the diagonal weight matrix depending on relation r, and $\mathbf{W}_0^{(l)}$ indicates a single self-connection of a special relation type to each node in the graph.

To learn the embeddings for the link prediction task, we used a margin-based loss function to minimize

$$\mathcal{L} = \sum_{\substack{(\mathbf{e_s},\mathbf{r},\mathbf{e_o})\in S \\ (\mathbf{e'_s},\mathbf{r},\mathbf{e'_o})\in S'_{(\mathbf{e_s},\mathbf{r},\mathbf{e_o})}}} [d(\mathbf{e_s}+\mathbf{r},\mathbf{e_o}) - d(\mathbf{e'_s}+\mathbf{r},\mathbf{e'_o}) + \gamma]_+, \tag{5.6}$$

where $[x]_+ = \max(0, x)$, $\mathbf{r} \in \mathbb{R}^d$, and $\gamma \in \{1, 2, 4\}$ is the margin hyper-parameter to define the margin distance. S is the set of positive pairs while S' is the set of negative pairs. Also, d($\cdot$) is the Euclidean distance function that generates lower values for positive triplets while it generates higher values for negative triplets as explained in the efforts by Bordes et al. (2012); LeCun et al. (2006).

The embeddings of entities will be initialized running convolution operation while the embeddings of relations ($\mathbf{r} \in \mathbb{R}^d$) will be initialized randomly as proposed by Glorot and Bengio (2010). After normalizing the embeddings, a small set of triplets will be sampled from the training set, and will serve as the training triplets. The parameters are then updated by taking a gradient step with constant learning rate.

5.4 Experimental Details

Our knowledge graph is essentially a directed graph with labeled edges from the set of 20 major relations as listed in Appendix A. Correspondingly, we totally collected 198,738 subject entities, 138,637 object entities, and 20 predicates as depth items in the partially filled data tensor as input for the TF experiments. Besides, we get totally 9,860,059 training examples with the combination of SemMedDB and UMLS predications for both methods. In the TF approach, we have two different scenarios: **binary tensor factorization (BTF)** where training relations are represented as 1s in the input tensor and **frequency count factorization (CTF)** where predications have their extraction frequency counts in the tensor cells. In the next subsection, we will describe the configurations of our models and the details of building the semantic knowledge graph for the GCN model.

5.4.1 Model Configurations & Details

In the TF experiments, we used an open source incomplete tensor approximation library, Splatt. The latent dimension k of a tensor to be factorized should be less than or equal to the minimum of tensor modes' dimensions. That is we run the experiments with the latent dimension of 20 which is the number of predicates used in the input tensor. We chose to leave the regularization parameters λ_A, λ_B, and λ_C as default (0.01) since tuning them did not yield any obvious gains in the experiments. Toward convergence based on the optimization equation in Eq. (5.4), a total of 100 iterations were used for incomplete tensor approximation.

In the GCN experiments, we employed 2-layer (l=2) convolution operation introduced by Schlichtkrull et al. To minimize the objective function, we selected the learning rate $\lambda \in \{0.001, 0.01\}$, the margin $\gamma \in \{1, 2, 4\}$ and the latent dimension $k = 50$. The optimal configuration was obtained using ADAM (Kingma and Ba, 2014) optimizer with a learning rate (λ) of 0.001, a margin (γ) of 1 , and a minibatch size of 32. We also applied early-stopping mechanism to avoid over-fitting. In this sense, training process is stopped early when the F-score performance on the validation set stops increasing for a particular number of consecutive epochs. We used early-stopping with 5-consecutive epoch criterion.

Complexity & Running Time Details: *For the TF experiment*, the complexity of the tensor factorization is $\mathcal{O}(kmnl)$ per iteration where k is the latent dimension, m is the number of rows, n is the number of columns, and l is the number of relations in the third dimension of the tensor. This is basically the total number of multiplications required for each cell in the given tensor (Kolda and Bader, 2009). The total running time including (identifying the best approximation of the input tensor and the evaluation of the test set with the found optimal threshold value on the validation set) was nearly three days for each scenario.

For the GCN experiment, the complexity of the convolution operation over our knowledge graph is $\mathcal{O}(nt^2)$ where n is number of nodes and t is the average node degree in the graph. The total running time of the GCN model (including fitting the model and evaluating the model with our test set) running on our GPU system was nearly two days. Overall, comparing all methods used in this dissertation, the most efficient method is the matrix completion because we have relatively fewer training examples than other methods and more importantly given it does not have to deal with other predicates of graph setup used in all other methods. On the other hand, the most expensive method in terms of the complexity is semantic patterns model as it needs the extraction of all the possible paths up to a particular length for each example pair.

5.4.2 Evaluation of Prediction Scores

To assess the predictive power of the models, we need to calculate performance metrics (precision, recall, and F-score). This is because we need a threshold t_{tf} above which we predict an entity pair as positive for the TF models. However, we predict a test pair as positive if the predicted value is less than the threshold t_{gcn} for the GCN approach; this is simply because we minimize the distance between the positive examples. Therefore, we generate a validation set containing randomly picked instances constituting 20% of positive examples and 20% of negative examples to explore optimum thresholds for each model.

We identified the threshold t_{tf} based on grid search over the validation dataset optimized for F-score with a small step size of 0.00001 spanning the range $[T^v_{min}, F^v_{max}]$ such that

$$T^v_{min} = \min(\{\hat{\mathcal{X}}_{i,j,z} : (i,j) \in T^v\}) \quad \text{and} \quad F^v_{max} = \max(\{\hat{\mathcal{X}}_{i,j,z} : (i,j) \in F^v\}),$$

where $\hat{\mathcal{X}}$ is the approximation of the input tensor $\mathcal{X}$ from Eq. (5.2), z is the depth index for the *treats* relation. Also, T^v and F^v represent ATs and FIs in the validation dataset respectively.

For the GCN experiments, we identified the threshold t_{gcn} with the same split of validation set and a higher step size of 0.1 spanning the range $[F^v_{min}, T^v_{max}]$. This is because we have a higher range of distance scores for the validation set.

5.5 Results & Evaluation

In this section, we first report the performance results of each approach. Then, we compare the performances with the NMF experiment reported in Chapter 4. In addition to individual results, we also present the performance scores of the ensemble of NMF, TF, and GCN models. Furthermore, we show the analysis of utilizing prior temporal data as we showed in Chapter 4 (Section 4.4.3) for each experiment. Subsequently, we present the experiments of error analysis with the test pairs predicted as either false positive (FP) or false negative (FN) in our best model.

5.5.1 Performance Evaluation of `repoDB` Pairs

In Table 5.1, we report the performance metrics of the models in both the TF and GCN experiments. Apparently, the CTF model has a better performance with the highest precision than both the BTF model and the GCN model. However, with the best GCN model, we get 2% lower F-score but achieve 7% higher recall performance than the best TF model. This is because around 7% recall difference cannot compensate 8% precision drop. On the other hand, binary tensor factorization (BTF) model yields the lowest scores for each performance measure. This is simply because the frequency count values provide the CTF model more predictive signal by reflecting the strength of the relations between the corresponding pairs compared to the BTF model. Ultimately, the tensor factorization approach (with the CTF model) surpassed the GCN approach with 2% higher F-score.

Table 5.1: Performance results of the `repoDB` test pairs with TF and GCN methods

Models	Precision	Recall	F-score
CTF model	**0.7617**	0.9292	**0.8371**
BTF model	0.6873	0.8184	0.7471
GCN model	0.6880	**0.9969**	0.8141

5.5.2 Ensemble of Models with Majority Voting

As we mentioned earlier in this section, we compare the best performance results of TF and GCN methods with our prior NMF based effort presented in the previous chapter.

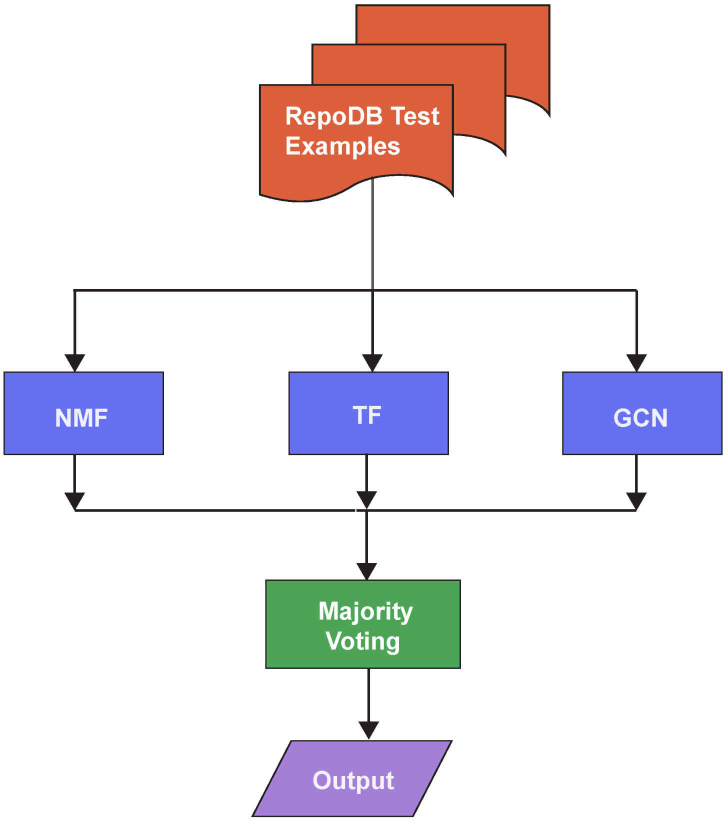

Figure 5.4: The ensemble model structure combining NMF, TF, and GCN models

Since we removed the all zero rows in the input matrix, we were left with 5,172 approved, 2,244 failed pairs in the NMF experiment. Unlike our prior work, there are 6,218 approved and 2,852 failed pairs available for the TF and GCN experiments as we mentioned in Section 5.2.2. This is because we utilize 20 relations instead of applying prediction only based on the *treats* relation.

Table 5.2: Predictive performance of TF, GCN, and NMF methods for drug respositioning

Approach	Precision	Recall	F-score
TF	0.7617	0.9292	**0.8371**
GCN	0.6880	**0.9969**	0.8141
NMF	**0.7882**	0.8823	0.8326

To make a fair comparison, we classified the missing test pairs in the NMF effort by random assignments. In Table 5.2, we present the best performance scores of the methods. Evidenced by compared performance scores, the TF model outperformed the NMF and GCN models based on the best F-score achieved. Interestingly, the

GCN model yielded the best recall and the lowest precision while the NMF model gained the best precision but has the lowest recall. Beyond that, we also checked the performances of the missing test pairs on TF and GCN models. We notice that almost all of the missing approved treatments are predicted as true positive (833 of 836) in the best GCN model. However, in the TF model, a total of 597 of them are predicted as true positive. Similarly, the TF model predicts 57 of 487 failed indications as true negative while the GCN model predicts only 6 of them as true negative. This indicates that TF and GCN methods are able to catch test pairs by means of indirect connections of the entities over essential relations (predicates) compared to the NMF approach. Thus, this outcome motivated us to build an ensemble model with the majority voting mechanism (Rokach, 2010) as depicted in Figure 5.4.

Table 5.3: The performance scores of the ensemble models for drug repositioning

Missing Pairs Vote	Precision	Recall	F-score
Same as GCN vote	**0.7529**	**0.9957**	**0.8575**
Random distribution	0.7502	0.9748	0.8479
Same as Tensor vote	0.7462	0.9477	0.8350

In the ensemble model experiment, the prediction of the missing pairs in the NMF model are assigned by three ways: random prediction, same prediction score as in the best GCN model, and same prediction score as in the best TF model. We report the performance results of ensemble models in Table 5.3. Unlike the individual performance of each approach, the ensemble approach yields superior prediction performance on the F-score metric. Here, the GCN approach helped the ensemble model to reach the best recall performance, while the NMF approach maintained the best precision performance. Thus, we achieved the best F-score performance (85%) with the ensemble model. Essentially, we realized that the predictions of our models in the ensemble are complementary for achieving the best trade-off between precision and recall with the highest F-score.

5.5.3 Performance Evaluation for *Prevents* & *Causes* Relations

Although we focus on the drug repositioning pairs in `repoDB`, we also report the performance scores of the pairs for the *prevents* and *causes* relations in Table 5.4. We aggregated totally 1,422 and 8,289 pairs as positive from UMLS Metathesaurus for *prevents* and *causes* relations respectively. We also generated same amount of negative examples as positive pairs as we introduced in Section 3.4.2. Similar to the repoDB experiment, we divided the obtained pairs into test (80% of the pairs) and validation (20% of the pairs) sets. When we compare the performance scores, the CTF model has a better F-score performance along with the highest precision scores for both relations. In the best case, we achieved 81% and 75% F-score performances for *prevents* and *causes* relations respectively.

Table 5.4: Performance results for *prevents* and *causes* relations with TF and GCN methods

	Prevents			Causes		
Models	Precision	Recall	F-score	Precision	Recall	F-score
CTF model	**0.8727**	0.7653	**0.8155**	**0.7734**	0.7314	**0.7518**
BTF model	0.7989	0.7398	0.7682	0.6648	0.7158	0.6894
GCN model	0.5998	**0.9972**	0.7323	0.6002	**0.9996**	0.7334

Moreover, the GCN model yielded an exceptional recall score and the worst precision performance for each relation. Interestingly, the BTF model achieved 3% better F-score than the GCN model for the *prevents* relation in consequence of having 20% higher precision score. On the other hand, the GCN model outperforms the BTF model with 5% higher F-score performance for the *causes* relation.

5.5.4 Prediction Role of Chronological Treatment Knowledge

As explained and performed earlier in Chapter 4, we conducted additional experiments to assess the effect of the exploiting prior therapeutic information on training for the drugs in A instances (ATs). When we check the positive test examples in repoDB, we notice that there are 516 unique drugs[†] treating more than one condition. Next, we aggregate the list of publication years of PubMed articles containing the corresponding TP pairs. Since treatment pairs can appear in multiple publications in different years, we collect the drug–disease AT pairs in $\mathcal{P}_{AT}$ where the corresponding publications for the disease are chronologically later than papers for other diseases the drug treats. Formally,

$$\mathcal{P}_{AT} = \{(d,c) : (d,c) \in AT,\ minDate(d,c) > maxDate(d,c')\ \forall_{c' \neq c}\, (d,c') \in AT\}, \quad (5.7)$$

[†] Note that we have more unique drugs than the NMF experiments because the missing test examples are present in this chapter.

where AT is the set of ATs and $maxDate(d, c)$ and $minDate(d, c)$ are functions that denote the latest and earliest publication dates discussing the corresponding AT pair (d, c), respectively. It is straightforward to see there is at most one pair $(d, c) \in \mathcal{P}_{AT}$ for any drug d. Ultimately, we obtained 168 pairs in the final test set $\mathcal{P}_{AT}$ when we examine the list of drug-disease pairs with their extraction years. After identifying the temporal test sets, we included the rest of the AT pairs as the prior knowledge for the temporal test set. Due to this, we had 513 additional training treatment pairs.

Table 5.5: Average prediction scores of temporal test pairs for drug repositioning with TF and GCN models

Models	w/o Temp.	w/ Temp.	Improve Rate
CTF model	0.3541	**0.6859**	48.38%
BTF model	0.0563	**0.0565**	**0.35%**
GCN model[‡]	7.2992	**5.1178**	29.88%

In Table 5.5, we demonstrate the average prediction scores of the temporal test examples for the TF and GCN experiments. Given the results show that the CTF model achieved the highest improvement (48%) compared to other models. However, the GCN model also yielded a solid improvement ($\approx$ 30%) while there was almost no impact of temporal knowledge in the BTF model. Clearly, the reported results indicate that using prior treatment knowledge of a drug contributes to predicting a recently discovered therapy for computational drug repositioning efforts.

5.5.5 Error Analysis with FP and FN Predictions on `repoDB`

To investigate the prediction error of our best model, we conduct error analysis experiments with false positive (FP) and false negative (FN) instances.

5.5.5.1 Medical Evaluation of FP Predictions

As we pointed out in Section 5.2.2, the negative test pairs are selected from clinical trials database. Thus, the clinical investigations implied a strong treatment possibility between the entities although the trials failed. To this end, we further examined FIs which were predicted as positive by each method. This filtration process resulted in a

[‡] In the GCN experiment, the model is optimized using the euclidean distance function that generates lower values for positive examples while it generates higher values for the negative examples

total of 731 FP test pairs[§]. Next, we ranked the pairs based on their prediction scores and picked 300 pairs containing the mixture of top 150 and last 150 pairs from the ranked list. Then, the pairs (sans their ranks) were independently reviewed by two practicing physicians (Dr. Tushi Singh and Dr. Romil Chadha) at the University of Kentucky hospital[¶]. We did not disclose to the reviewers that all pairs were FIs given we wanted to see which one of those do the physicians think are actually plausible therapies. The pairs were rated based on the scale in Table 5.6.

Table 5.6: The rating scale for the treatment plausibility rating

Description	Rating Score
Strongly disagree	1
Disagree	2
Neutral	3
Agree	4
Strongly agree	5

After they completed independent rating, we found that the ratings of 255 pairs are either matched or have a difference of one positions. The remaining 45 pairs have at least two position differences. Next, the raters resolved the conflicts of these 45 pairs to come to a mutually agreed upon rating after face to face discussions.

Table 5.7: Average physician rating scores of biomedical plausibility for drug-disease pairs

	Physician-1			Physician-2		
Average Cases	Top-150	Last-150	All	Top-150	Last-150	All
Before agreement	3.6	3.33	3.48	3.66	3.41	3.53
After agreement	3.65	3.4	3.52	3.62	3.41	3.51

In Table 5.7, we present the average scores of 300 FP pairs rated by medical domain experts. The difference of average ratings for all pairs dropped to 0.01 from 0.05 after two physicians resolved their disagreements. Given the average rating scores, it is clear that the physicians ratings align with our model's predictions, since the average score is around 3.5 which is higher than the neutral score and lower than

[§] In the matrix completion experiment, we had 940 FPs while in TF and GCN experiments we had 1,425 and 2,280 (out of 2,282) FPs.

[¶] Since finding specific domain experts on each test pair is highly time consuming for the evaluation, we just collaborated with MDs who are working in internal medicine department. However, evaluation of the test cases with specific domain experts may yield better evaluations.

the agree score. Furthermore, we found that there is around 6% difference between the rating scores of the top and last pairs. Thus, we show that our model scores highlight the relative plausibilities of potential therapeutic connections as rated by the physicians. More importantly, we note that the manual assigned average rating of the FP pairs is above ‘neutral’. This is because the examples predicted as FPs are essentially failed clinical trials in `repoDB`. Therefore, those were assumed plausible enough for the medical investigations by researchers.

5.5.5.2 FN Analysis with Semantic Patterns

Unlike the FP pairs, we do not have many test pairs predicted as FNs. Therefore, we picked three top and three bottom pairs based on their prediction scores from each method used in the ensemble model (a total of 18 FN pairs). To investigate the explanations for FNs, we checked the semantic patterns between entities in the pairs as discussed in Chapter 3. Basically, we examine whether the selected FN pairs are connected more through the graph patterns that correlate with treatment relations in comparison with patterns that correlate with the negative class in the treatment prediction model.

Table 5.8: The average number of matching top semantic patterns for FN pairs

Test Case	Negative Patterns		Positive Patterns	
	Top-1K	Top-10K	Top-1K	Top-10K
Top 9 Pairs	430	3483	148	1116
Bottom 9 Pairs	551	4200	178	1291
All 18 Pairs	**491**	**3842**	163	1204

We report the number of matching semantic patterns of the FN pairs in top negative and positive discriminative patterns as shown in Table 5.8. It is clear that the pairs have more negative patterns connecting them than the positive ones. Thus, having more indirect associations common in the negative class justifies the FN predictions of the given test pairs. Since top pairs have fewer matches with the positive patterns, we can infer that the ranking is reasonable although bottom pairs have more negative patterns matched.

5.6 Conclusion

Computational drug repositioning is a practical way of expediting the drug development process. In this chapter, we applied tensor factorization and graph convolutional networks with SemMedDB and UMLS repositories to classify drug repositioning pairs in repoDB. Moreover, we compared the results with our previous NMF based effort and built an ensemble method using NMF, TF, and GCN approaches. We achieved **75**% precision, **99**% recall, and **85**% F-score. With the ensemble method, we just have 3% loss of precision from the best precision case and gained 2% in F-score compared to the single model giving the best F-score. This is because we did not have any significant recall drop from the best individual model giving the highest recall (only 0.12% loss). We conducted error analysis with FP and FN pairs. In this regard, we justified the FP predictions of our ensemble model by analyzing the plausibility ratings assigned by medical experts. Similarly, we demonstrated that FN pair predictions had more semantic patterns matching with top negative patterns (correlating with the negative class) obtained from the treatment relation prediction task in (Bakal et al., 2018).

Chapter 6 Conclusion and Future Work

Discovering new potential treatment options for medical conditions which cause human disease burden is an essential goal of medical research. However, the excessive time and financial cost of new drug development are nontrivial obstacles for pharmaceutics and biomedical communities. Therefore, computational efforts are used to provide plausible drug candidates for drug repositioning research, which is an efficient way to complement the traditional drug development strategies. In this dissertation, we employed supervised machine learning methods that predict approved and failed drug repositioning pairs with reasonably high predictive performance. In the rest of this chapter, we present the summary of the contributions to the field and the limitations of the efforts we demonstrated in the earlier chapters.

6.1 Summary of Contributions

In this dissertation, we presented different machine learning methods for relation prediction task in drug repositioning research. We list the main contributions in this research below:

Semantic Patterns over Knowledge Graphs. In Chapter 3, we employed semantic graph patterns extracted from biomedical knowledge graphs connecting pairs of candidate entities as a set of features to predict treatment and causative relations between them. To that end, we built various models with logistic regression and decision tree classifiers for the prediction task. We report that semantic patterns over knowledge graphs hold great promise for global relation prediction in biomedicine. Moreover, we analyzed the top patterns informed by model coefficients and demonstrated their interpretability for gaining insights into the prediction process. Also, we examined false positives with high probability outputs assigned by our model based on the inputs from practicing physicians.

Non-negative Matrix Factorization. Matrix completion through NMF based low-rank approximation is an effective method for computational drug repositioning based on datasets of previously approved drugs and corresponding indications. In Chapter 4, we presented the first effort for repositioning that directly tests against `repoDB` instances. By using hand-curated drug–disease indications from the UMLS Metathesaurus and automatically extracted relations from the SemMedDB knowledge base, we employed non-negative matrix factorization (NMF) methods to recover `repoDB` positive indications.

Tensor Factorization & Graph Convolutional Neural Networks. Knowledge base completion is a popular way of predicting missing links between entities. In biomedical domain, drug repositioning is a special case of knowledge base completion by exploring potential new treatment connections between biomedical entities. In Chapter 5, we built tensor factorization (TF) and graph convolutional neural network (GCN) models exploiting UMLS and SemMedDB knowledge bases to predict drug repositioning test pairs in `repoDB`. Furthermore, we compared the performance scores of the TF and GCN models with the NMF model and generated an ensemble model using majority voting mechanism with them. We achieved the best prediction performance with the ensemble model indicating that all three models have complementary traits.

6.2 Limitations and Future Work

There are some limitations of the models discussed in this manuscript. For the semantic pattern model explained in Chapter 3, our training examples must be connected in the SemMedDB graph with at least one path. Otherwise, the feature vector will be a zero vector and will have no information for the prediction task. Another issue with the LR model is the need to extract the paths connecting entities of length $\leq k$. Even though we handled $k = 3$ using a straightforward heuristic, we are not aware of a simple way to do the same for $k > 3$. A final caveat is that each predicate needs to have a separate binary model.

For the NMF model in Chapter 4, matrix completion methods cannot populate a row which does not have at least one nonzero entry (shared contextual information with other rows). This means, in our case, a drug for which we do not have at least one known treatment relation cannot be linked to new indications with NMF. However, we overcame this issue by moving from matrices to tensors with additional relations between entities connected with other biomedical predicates that are listed in Appendix A. Consequently, for the TF and GCN models explained in Chapter 5, we have lower precision scores compared to recall scores (over 90% for both). This is because the examples predicted as FPs are essentially failed cases in `repoDB`. That means that those failed indications were assumed plausible enough for researchers to launch clinical investigations.

We believe that there are some future directions as follows:

1. For the count based models in Chapters 4 and 5, we simply employed raw frequencies of biomedical predications in SemMedDB instead of standardizing counts using well-known methods (e.g., mean centering, min-max scaling, log transformation). Thus, using raw frequencies may have lead to potential ill-conditioning that needs to be countered with appropriate pre-processing and/or using more sophisticated approaches. We believe that these experiments will be a part of future work for our NMF and TF efforts.

2. In this dissertation, we have focused on predicting general treatment pairs and especially drug repositioning pairs in `repoDB`. However, in general scientists are often looking at a particular disease that they want to treat. Hence, for disease specific computational drug repositioning efforts, integration of disease specific information would be helpful to increase the accuracy of the prediction for disease specific pairs.

3. In our matrix completion method, we obtained fairly reasonable predictive performance. In light of this outcome, we can exploit this predictive power by applying the NMF method to slices corresponding to essential predicates (e.g. *stimulates* and *prevents*) from the 3D knowledge tensor that we were factorizing in the TF method. After obtaining plausible new knowledge with the matrix completion approach, we can add corresponding new links to update our input tensor so as to have more existing knowledge (training data) to analyze the predictive power of the TF approach fed by NMF models.

Appendix A : 20 Major Predicates

- TREATS
- INTERACTS_WITH
- AFFECTS
- COEXISTS_WITH
- ASSOCIATED_WITH
- CAUSES
- INHIBITS
- STIMULATES
- AUGMENTS
- DISRUPTS
- PREDISPOSES
- PREVENTS
- ISA
- ADMINISTERED_TO
- NEG_AFFECTS
- NEG_TREATS
- PRODUCES
- USES
- OCCURS_IN
- SAME_AS

Appendix B : Abbreviations

ALS Alternating Least Squares. 57

AT Approved Treatment. 53, 54, 65

BTF Binary Tensor Factorization. 60, 62

CDR Computational Drug Repositioning. 7, 35, 36

CNN Convolutional Neural Networks. 11

CP CANDECOM/PARAFAC – CANonical DECOMposition/PARAlel FACtors. 55

CTF Frequency-count Tensor Factorization. 60

CUI Concept Unique Identifiers. 37

DR Drug Repositioning. 35

DT Decision Tree. 4, 20

FDA Food and Drug Administration. 26, 35

FI Failed Indication. 53, 54

FN False Negative. 12, 66

FP False Positive. 12, 31, 66

GCN Graph Convolutional Neural Network. 7, 11, 53, 58, 71

GPU Graphics Processing Unit. 10, 61

LBD Literature Based Discovery. 6

LR Logistic Regression. 4, 17

MeSH Medical Subject Heading. 35

ML Machine Learning. 1

NLM National Library of Medicine. 14, 35

NLP Natural Language Processing. 1, 6, 36

NMF Non-negative Matrix Factorization. 4, 37, 40, 71

NN Neural Networks. 10

NTF Non-negative Tensor Factorization. 55

SemMedDB Semantic MEDLINE Database. 37

SVD Singular Value Decomposition. 10

SVM Support Vector Machine. 9, 20

TF Tensor Factorization. 53, 71

TP True Positive. 12

UMLS Unified Medical Language System. 37

Bibliography

[1] A. B. Abacha and P. Zweigenbaum. "Automatic extraction of semantic relations between medical entities: a rule based approach". In: *Journal of biomedical semantics* 2.5 (2011), S4.

[2] E. Acar, D. M. Dunlavy, and T. G. Kolda. "A scalable optimization approach for fitting canonical tensor decompositions". In: *Journal of Chemometrics* 25.2 (2011), pp. 67–86.

[3] C. Andronis, A. Sharma, V. Virvilis, S. Deftereos, and A. Persidis. "Literature mining, ontologies and information visualization for drug repurposing". In: *Briefings in bioinformatics* 12.4 (2011), pp. 357–368.

[4] G. Bakal and R. Kavuluru. "Predicting Treatment Relations with Semantic Patterns over Biomedical Knowledge Graphs". In: *International Conference on Mining Intelligence and Knowledge Exploration*. Springer. 2015, pp. 586–596.

[5] G. Bakal, P. Talari, E. V. Kakani, and R. Kavuluru. "Exploiting semantic patterns over biomedical knowledge graphs for predicting treatment and causative relations". In: *Journal of biomedical informatics* 82 (2018), pp. 189–199.

[6] R. A. van den Berg, H. C. Hoefsloot, J. A. Westerhuis, A. K. Smilde, and M. J. van der Werf. "Centering, scaling, and transformations: improving the biological information content of metabolomics data". In: *BMC genomics* 7.1 (2006), p. 142.

[7] A. Bordes, X. Glorot, J. Weston, and Y. Bengio. "A semantic matching energy function for learning with multi-relational data". In: *Machine Learning* (2012), pp. 1–27.

[8] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. "Translating embeddings for modeling multi-relational data". In: *Advances in neural information processing systems*. 2013, pp. 2787–2795.

[9] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[10] A. S. Brown and C. J. Patel. "A standard database for drug repositioning". In: *Scientific Data* 4 (2017), p. 170029.

[11] D. Cameron, O. Bodenreider, H. Yalamanchili, T. Danh, S. Vallabhaneni, K. Thirunarayan, A. P. Sheth, and T. C. Rindflesch. “A graph-based recovery and decomposition of Swanson’s hypothesis using semantic predications”. In: *Journal of biomedical informatics* 46.2 (2013), pp. 238–251.

[12] D. Cameron, R. Kavuluru, T. C. Rindflesch, A. P. Sheth, K. Thirunarayan, and O. Bodenreider. “Context-Driven Automatic Subgraph Creation for Literature-Based Discovery”. In: *Journal of biomedical informatics* 54 (2015), pp. 141–157.

[13] D. Cameron, R. Kavuluru, T. C. Rindflesch, A. P. Sheth, K. Thirunarayan, and O. Bodenreider. “Context-driven automatic subgraph creation for literature-based discovery”. In: *Journal of biomedical informatics* 54 (2015), pp. 141–157.

[14] L. Cheng, H. Lin, F. Zhou, Z. Yang, and J. Wang. “Enhancing the accuracy of knowledge discovery: a supervised learning method”. In: *BMC bioinformatics* 15.12 (2014), S9.

[15] W.-S. Chin, Y. Zhuang, Y.-C. Juan, and C.-J. Lin. “A fast parallel stochastic gradient method for matrix factorization in shared memory systems”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 6.1 (2015), p. 2.

[16] A. Cichocki and R. Zdunek. “Multilayer nonnegative matrix factorisation”. In: *Electronics Letters* 42.16 (2006), pp. 947–948.

[17] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation.* John Wiley & Sons, 2009.

[18] T. Cohen, D. Widdows, C. Stephan, R. Zinner, J. Kim, T. Rindflesch, and P. Davies. “Predicting high-throughput screening results with scalable literature-based discovery methods”. In: *CPT: pharmacometrics & systems pharmacology* 3.10 (2014), e140.

[19] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. “Indexing by latent semantic analysis”. In: *Journal of the American society for information science* 41.6 (1990), pp. 391–407.

[20] J. A. DiMasi, H. G. Grabowski, and R. W. Hansen. “Innovation in the pharmaceutical industry: new estimates of R&D costs”. In: *Journal of health economics* 47 (2016), pp. 20–33.

[21] J. T. Dudley, M. Sirota, M. Shenoy, R. K. Pai, S. Roedder, A. P. Chiang, A. A. Morgan, M. M. Sarwal, P. J. Pasricha, and A. J. Butte. "Computational repositioning of the anticonvulsant topiramate for inflammatory bowel disease". In: *Science translational medicine* 3.96 (2011), 96ra76–96ra76.

[22] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. "Convolutional networks on graphs for learning molecular fingerprints". In: *Advances in neural information processing systems*. 2015, pp. 2224–2232.

[23] R. de Fréin, K. Drakakis, S. Rickard, and A. Cichocki. "Analysis of financial data using non-negative matrix factorization". In: *International Mathematical Forum*. Vol. 3. 38. Journals of Hikari Ltd. 2008, pp. 1853–1870.

[24] K. Fundel, R. Küffner, and R. Zimmer. "RelEx - Relation extraction using dependency parse trees". In: *Bioinformatics* 23.3 (2007), pp. 365–371.

[25] X. Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. 2010, pp. 249–256.

[26] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT Press, 2016.

[27] A. Gottlieb, G. Y. Stein, E. Ruppin, and R. Sharan. "PREDICT: a method for inferring novel drug indications with application to personalized medicine". In: *Molecular systems biology* 7.1 (2011), p. 496.

[28] D. Jannach, P. Resnick, A. Tuzhilin, and M. Zanker. "Recommender systems — beyond matrix completion". In: *Communications of the ACM* 59.11 (2016), pp. 94–102.

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

[30] R. Kavuluru, C. Thomas, A. P. Sheth, V. Chan, W. Wang, A. Smith, A. Soto, and A. Walters. "An up-to-date knowledge-based literature search and exploration framework for focused bioscience domains". In: *Proc. of the 2nd ACM SIGHIT Health Informatics Symposium*. ACM. 2012, pp. 275–284.

[31] H. Kilicoglu, D. Shin, M. Fiszman, G. Rosemblat, and T. C. Rindflesch. “SemMedDB: a PubMed-scale repository of biomedical semantic predications”. In: *Bioinformatics* 28.23 (2012), pp. 3158–3160.

[32] S. Kim, H. Liu, L. Yeganova, and W. J. Wilbur. “Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach”. In: *Journal of biomedical informatics* 55 (2015), pp. 23–30.

[33] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations (ICLR)*. 2014.

[34] T. N. Kipf and M. Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In: *International Conference on Learning Representations (ICLR)*. 2017.

[35] D. G. Kleinbaum and M. Klein. *Logistic Regression: A Self-Learning Text*. Statistics for Biology and Health. Springer New York, 2010.

[36] T. G. Kolda and B. W. Bader. “Tensor decompositions and applications”. In: *SIAM review* 51.3 (2009), pp. 455–500.

[37] M. Koudelka and D. J. Dorsey. “A Modular NMF Matching Algorithm for Radiation Spectra”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016, pp. 18–23.

[38] D. Kuang, J. Choo, and H. Park. “Nonnegative matrix factorization for interactive topic modeling and document clustering”. In: *Partitional Clustering Algorithms*. Springer, 2015, pp. 215–243.

[39] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. “A tutorial on energy-based learning”. In: *Predicting structured data* 1 (2006).

[40] J. Li and Z. Lu. “A new method for computational drug repositioning using drug pairwise similarity”. In: *Bioinformatics and Biomedicine (BIBM), 2012 IEEE International Conference On*. IEEE. 2012, pp. 1–4.

[41] J. Li and Z. Lu. “Pathway-based drug repositioning using causal inference”. In: *BMC bioinformatics* 14.16 (2013), S3.

[42] J. Li, S. Zheng, B. Chen, A. J. Butte, S. J. Swamidass, and Z. Lu. “A survey of current trends in computational drug repositioning”. In: *Briefings in bioinformatics* 17.1 (2016), pp. 2–12.

[43] C.-J. Lin. “On the convergence of multiplicative update algorithms for nonnegative matrix factorization”. In: *IEEE Transactions on Neural Networks* 18.6 (2007), pp. 1589–1596.

[44] Y. Liu, R. Bill, M. Fiszman, T. Rindflesch, T. Pedersen, G. B. Melton, and S. V. Pakhomov. “Using SemRep to label semantic relations extracted from clinical text”. In: *AMIA annual symposium proceedings*. Vol. 2012. American Medical Informatics Association. 2012, p. 587.

[45] Z. Lu. “PubMed and beyond: a survey of web tools for searching biomedical literature”. In: *Database: the journal of biological databases and curation* 2011 (2011).

[46] Y. Luo, F. Wang, and P. Szolovits. “Tensor factorization toward precision medicine”. In: *Briefings in bioinformatics* 18.3 (2016), pp. 511–514.

[47] M. Mintz, S. Bills, R. Snow, and D. Jurafsky. “Distant supervision for relation extraction without labeled data”. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. ACL. 2009, pp. 1003–1011.

[48] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[49] A. Nagaraj, Q. Wang, P. Joseph, C. Zheng, Y. Chen, O. Kovalenko, S. Singh, A. Armstrong, K. Resnick, K. Zanotti, et al. “Using a novel computational drug-repositioning approach (DrugPredict) to rapidly identify potent drug candidates for cancer treatment”. In: *Oncogene* 37.3 (2018), p. 403.

[50] National Library of Medicine. *Current Hierarchy of UMLS Predicates*. `http://www.nlm.nih.gov/research/umls/META3_current_relations.html`. 2003.

[51] National Library of Medicine. *Current Hierarchy of UMLS Semantic Types*. `http://www.nlm.nih.gov/research/umls/META3_current_semantic_types.html`. 2003.

[52] National Library of Medicine. *Semantic MEDLINE Database*. `http://skr3.nlm.nih.gov/SemMedDB/`. 2016.

[53] National Library of Medicine. *SemRep - NLM’s Semantic Predication Extraction Program*. `http://semrep.nlm.nih.gov`. 2013.

[54] National Library of Medicine. *Unified Medical Language System Reference Manual*. `http://www.ncbi.nlm.nih.gov/books/NBK9676/`. 2009.

[55] M. Nickel, V. Tresp, and H.-P. Kriegel. “A Three-Way Model for Collective Learning on Multi-Relational Data.” In: *ICML*. Vol. 11. 2011, pp. 809–816.

[56] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. “Scikit-learn: Machine learning in Python”. In: *Journal of Machine Learning Research* 12.Oct (2011), pp. 2825–2830.

[57] F. Provost and R. Kohavi. “Glossary of terms”. In: *Journal of Machine Learning* 30.2-3 (1998), pp. 271–274.

[58] S. Riedel, L. Yao, and A. McCallum. “Modeling relations and their mentions without labeled text”. In: *Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 148–163.

[59] T. C. Rindflesch and M. Fiszman. “The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text”. In: *Journal of Biomedical Informatics* 36.6 (2003), pp. 462–477.

[60] A. Ritter, L. Zettlemoyer, O. Etzioni, et al. “Modeling missing data in distant supervision for information extraction”. In: *Transactions of the Association for Computational Linguistics* 1 (2013), pp. 367–378.

[61] L. Rokach. “Ensemble-based classifiers”. In: *Artificial Intelligence Review* 33.1 (2010), pp. 1–39.

[62] S. Roy, R. Homayouni, M. W. Berry, and A. A. Puretskiy. “Nonnegative tensor factorization of biomedical literature for analysis of genomic data”. In: *Data Mining for Service*. Springer, 2014, pp. 97–110.

[63] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling. “Modeling relational data with graph convolutional networks”. In: *European Semantic Web Conference*. Springer. 2018, pp. 593–607.

[64] I. Segura-Bedmar, P. Martinez, and C. de Pablo-Sánchez. “Using a shallow linguistic kernel for drug–drug interaction extraction”. In: *Journal of biomedical informatics* 44.5 (2011), pp. 789–804.

[65] S. Smith, N. Ravindran, N. D. Sidiropoulos, and G. Karypis. “SPLATT: Efficient and parallel sparse tensor-matrix multiplication”. In: *Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International*. IEEE. 2015, pp. 61–70.

[66] R. Socher, D. Chen, C. D. Manning, and A. Ng. “Reasoning with neural tensor networks for knowledge base completion”. In: *Advances in neural information processing systems*. 2013, pp. 926–934.

[67] R. S. Stafford. “Regulating off-label drug use—rethinking the role of the FDA”. In: *New England Journal of Medicine* 358.14 (2008), pp. 1427–1429.

[68] M. Surdeanu, J. Tibshirani, R. Nallapati, and C. D. Manning. “Multi-instance multi-label learning for relation extraction”. In: *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 2012, pp. 455–465.

[69] A. Tasneem, L. Aberle, H. Ananth, S. Chakraborty, K. Chiswell, B. J. McCourt, and R. Pietrobon. “The database for aggregate analysis of ClinicalTrials. gov (AACT) and subsequent regrouping by clinical specialty”. In: *PloS one* 7.3 (2012), e33677.

[70] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard. “Complex embeddings for simple link prediction”. In: *International Conference on Machine Learning*. 2016, pp. 2071–2080.

[71] O. Ursu, J. Holmes, J. Knockel, C. G. Bologa, J. J. Yang, S. L. Mathias, S. J. Nelson, and T. I. Oprea. “DrugCentral: online drug compendium”. In: *Nucleic acids research* 45.D1 (2017), pp. D932–D939.

[72] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. “Graph Attention Networks”. In: *International Conference on Learning Representations* (2018).

[73] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos. “Class imbalance, redux”. In: *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE. 2011, pp. 754–763.

[74] Y.-X. Wang and Y.-J. Zhang. “Nonnegative matrix factorization: A comprehensive review”. In: *IEEE Transactions on Knowledge and Data Engineering* 25.6 (2013), pp. 1336–1353.

[75] Y. Wang, R. Chen, J. Ghosh, J. C. Denny, A. Kho, Y. Chen, B. A. Malin, and J. Sun. “Rubik: Knowledge guided tensor factorization and completion for health data analytics”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2015, pp. 1265–1274.

[76] Z. Wang, J. Zhang, J. Feng, and Z. Chen. “Knowledge graph embedding by translating on hyperplanes”. In: *Twenty-Eighth AAAI conference on artificial intelligence*. 2014.

[77] B. Wilkowski, M. Fiszman, C. M. Miller, D. Hristovski, S. Arabandi, G. Rosemblat, and T. C. Rindflesch. “Graph-based methods for discovery browsing with semantic predications”. In: *AMIA annual symposium proceedings*. Vol. 2011. American Medical Informatics Association. 2011, p. 1514.

[78] T. E. Workman, M. Fiszman, M. J. Cairelli, D. Nahl, and T. C. Rindflesch. “Spark, an application based on Serendipitous Knowledge Discovery”. In: *Journal of biomedical informatics* 60 (2016), pp. 23–37.

[79] H. Xu, M. C. Aldrich, Q. Chen, H. Liu, N. B. Peterson, Q. Dai, M. Levy, A. Shah, X. Han, X. Ruan, et al. “Validating drug repurposing signals using electronic health records: a case study of metformin associated with reduced cancer mortality”. In: *Journal of the American Medical Informatics Association* (2014), amiajnl–2014.

[80] R. Xu, A. Morgan, A. K. Das, and A. Garber. “Investigation of unsupervised pattern learning techniques for bootstrap construction of a medical treatment lexicon”. In: *Proceedings of the workshop on current trends in biomedical natural language processing*. Association for Computational Linguistics. 2009, pp. 63–70.

[81] W. Xu, R. Hoffmann, L. Zhao, and R. Grishman. “Filling Knowledge Base Gaps for Distant Supervision of Relation Extraction.” In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. ACL. 2013, pp. 665–670.

[82] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng. “Embedding entities and relations for learning and inference in knowledge bases”. In: *arXiv preprint arXiv:1412.6575* (2014).

[83] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. “Deconvolutional networks”. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE. 2010, pp. 2528–2535.

[84] H. Zhang, M. Fiszman, D. Shin, C. M. Miller, G. Rosemblat, and T. C. Rindflesch. “Degree centrality for semantic abstraction summarization of therapeutic studies”. In: *Journal of biomedical informatics* 44.5 (2011), pp. 830–838.

[85] P. Zhang, F. Wang, and J. Hu. “Towards drug repositioning: a unified computational framework for integrating multiple aspects of drug similarity and disease similarity”. In: *AMIA Annual Symposium Proceedings*. Vol. 2014. American Medical Informatics Association. 2014, p. 1258.

[86] R. Zhang, M. J. Cairelli, M. Fiszman, H. Kilicoglu, T. C. Rindflesch, S. V. Pakhomov, and G. B. Melton. “Exploiting Literature-derived Knowledge and Semantics to Identify Potential Prostate Cancer Drugs”. In: *Cancer Informatics* 13 (2014), pp. 103–111.

[87] R. Zhang, M. J. Cairelli, M. Fiszman, G. Rosemblat, H. Kilicoglu, T. C. Rindflesch, S. V. Pakhomov, and G. B. Melton. “Using semantic predications to uncover drug–drug interactions in clinical data”. In: *Journal of biomedical informatics* 49 (2014), pp. 134–147.

[88] Q. Zhu, C. Tao, F. Shen, and C. Chute. “Exploring the pharmacogenomics knowledge base (pharmgkb) for repositioning breast cancer drugs by leveraging Web ontology language (owl) and cheminformatics approaches”. In: *19th Pacific Symposium on Biocomputing, PSB 2014*. 2014, pp. 172–182.

[89] M. Zitnik, M. Agrawal, and J. Leskovec. “Modeling polypharmacy side effects with graph convolutional networks”. In: *Bioinformatics* 34.13 (2018), pp. i457–i466.

Vita

Name

Mehmet Gokhan Bakal

Education

- 2012–2014 M.S. in Computer Science at University of Kentucky, Lexington, Kentucky
- 2005–2009 B.S. in Computer Engineering at Trakya University, Edirne, Turkey

Experience

- 2009-2011 Software Engineer & Certified OpenText Consultant, DDI Technology, Istanbul, Turkey
- Summer 2008 Software Engineer Intern, Naryaz Computer and Software Ltd., Istanbul, Turkey
- Summer 2008 Software Engineer Intern, BILGEADAM - IT Training & Consulting Company, Istanbul, Turkey
- Summer 2007 Software Engineer Intern, Naryaz Computer and Software Ltd., Istanbul, Turkey

Awards

- 2011-2019, A scholarship program for graduate education abroad provided by Turkish government

Publications

- **Gokhan Bakal**, Halil Kilicoglu, and Ramakanth Kavuluru. Non-Negative Matrix Factorization for Drug Repositioning: Experiments with the repoDB Dataset. To be appeared in the Proceeding of the AMIA 2019 Annual Symposium.

- **Bakal G**, Talari P, Kakani EV, Kavuluru R. Exploiting semantic patterns over biomedical knowledge graphs for predicting treatment and causative relations. Journal of biomedical informatics. 82:189-99; 2018.
- **Bakal G**, Kavuluru R. On quantifying diffusion of health information on Twitter. In IEEE EMBS International Conference on Biomedical & Health Informatics (BHI); 2017.
- **Bakal G**, Wallace S, Kavuluru R. On the Predictive Potential of Graph Patterns for Biomedical Relation Extraction. In American Medical Informatics Association (AMIA) annual symposium; 2016.
- **Bakal G**, Kavuluru R. Predicting treatment relations with semantic patterns over biomedical knowledge graphs. In International Conference on Mining Intelligence and Knowledge Exploration; 2015.

Academic Service

- Reviewed papers for AMIA 2017, 2018, and 2019.