Large-Scale Structural and Textual Similarity-Based Mining of Knowledge Graph to Predict Drug-Drug Interactions

Ibrahim Abdelaziz[a,*], Achille Fokoue[b], Oktie Hassanzadeh[b], Ping Zhang[b], Mohammad Sadoghi[c]

[a] *King Abdullah University of Science & Technology, KSA*
[b] *IBM T.J. Watson Research Center, Yorktown Heights, NY, USA*
[c] *Department of Computer Science, Purdue University, West Lafayette, IN, USA*

Abstract

Drug-Drug Interactions (DDIs) are a major cause of preventable Adverse Drug Reactions (ADRs), causing a significant burden on the patients' health and the healthcare system. It is widely known that clinical studies cannot sufficiently and accurately identify DDIs for new drugs before they are made available on the market. In addition, existing public and proprietary sources of DDI information are known to be incomplete and/or inaccurate and so not reliable. As a result, there is an emerging body of research on in-silico prediction of drug-drug interactions. In this paper, we present Tiresias, a large-scale similarity-based framework that predicts DDIs through link prediction.Tiresias takes in various sources of drug-related data and knowledge as inputs, and provides DDI predictions as outputs. The process starts with semantic integration of the input data that results in a knowledge graph describing drug attributes and relationships with various related entities such as enzymes, chemical structures, and pathways. The knowledge graph is then used to compute several similarity measures between all the drugs in a scalable and distributed framework. In particular, Tiresias utilizes two classes of features in a knowledge graph: local and global features. Local features are derived from the information directly associated to each drug (i.e., one hop away) while global features are learnt by minimizing a global loss function that considers the complete structure of the knowledge graph. The resulting similarity metrics are used to build features for a large-scale logistic regression model to predict potential DDIs. We highlight the novelty of our proposed Tiresias and perform thorough evaluation of the quality of the predictions. The results show the effectiveness of Tiresias in both predicting new interactions among existing drugs as well as newly developed drugs.

Keywords: Drug Interaction, Similarity-Based, Link Prediction

1. Introduction

Adverse drug reactions (ADRs) is now becoming the 4th leading cause of deaths in United States surpassing complex diseases such as diabetes, pneumonia, and AIDS [1]. Over two million ADRs are being reported in U.S. annually that sadly results in 100,000 loss of life every year. Furthermore, a significant resource of $136 billion is dedicated to treat complications arised due to ADRs. In fact, the cost of care for attempting to reverse ADRs symptoms is higher than the cost of care for both diabetic and cardiovascular combined. More importantly, a detailed analysis of ADR incidents reveals that approximately 3 to 5% of all in-hospital medication errors are due to "preventable" drug-drug interactions (DDIs) [1].

Therefore, a natural question arises as to why so many preventable DDIs continues to plaque patients and the healthcare system as a whole, the answer is twofold. First, despite the advances made in drug development and safety, clinical trails often fail to reveal rare toxicity of certain drugs given the limited size and length of these studies. For instance, an average typical trail for any drug is limited to only 1,500 patients for rather a short period of time. Therefore, they fail to show the actual impacts of the drug once offered to millions of patients for much longer period of time. These concerns are further exacerbated as it is well known that adverse reaction increases exponentially when taking four or more drugs simultaneously [1]. Consequently, the rare toxicity of newly developed drugs cannot be established until after the drug becomes widely available in the market. Second, to make the matter worse, healthcare providers often fail to report ADRs because they have a misconception that all severe adverse reactions are already known when a drug is brought to the market [1].

Recently, there is a growing interest in computationally predicting potential DDIs [2, 3, 4, 5, 6, 7, 8]. These approaches are broadly classified as either similarity (e.g., [2, 6, 7]) or feature-based (e.g., [3]) DDI predication methods. There are a set of significant challenges and shortcomings that that are mostly overlooked by prior work. We summarize each of these limitations as follows:

Problem 1: Inability to make predictions for newly developed drugs. Prior work either (*i*) are fundamentally unable

[*] Corresponding author

Email addresses: `ibrahim.abdelaziz@kaust.edu.sa` (Ibrahim Abdelaziz), `achille@us.ibm.com` (Achille Fokoue), `hassanzadeh@us.ibm.com` (Oktie Hassanzadeh), `pzhang@us.ibm.com` (Ping Zhang), `msadoghi@purdue.edu` (Mohammad Sadoghi) to make predictions for newly developed drugs (i.e., drugs for which no or very limited information about interacting drugs is available) [8] or (*ii*) could conceptually predict drugs interacting with a new drug, but have not been tested for this scenario [2, 7]. Similarity-based approaches (e.g. [2, 7]) can clearly be applied to drugs without any known interacting drugs. However, in commonly carried 10 fold cross validation evaluation, prior work using similarity-based approaches have hidden drug-drug interaction associations and not drugs. Thus, the large majority of drugs used at testing are also known during the training phase, which is an inappropriate evaluation strategy to simulate the introduction of a newly developed drug. In our experimental evaluation, we show that the prediction quality of the basic similarity-based approaches drops noticeably when hiding drugs instead of drug-drug associations.

Problem 2: Ignoring the skewed distribution of interacting drug pairs. Most prior work [2, 6, 7] assume *a priori* a balanced distribution of interacting drug pairs at training or at testing. There is no reason to believe that the prevalence of pairs of interacting drugs in the set of all the drug pairs is close to 50% (often falsely assumed in past studies).

Problem 3: Discarding many relevant data sources and incompleteness of similarity measures. Existing techniques [3, 2, 7, 6] have relied on a limited number of data sources (primarily DrugBank) for creating drug similarity measures. Since various data sources provide only partial information about a subset of drugs of interest, the resulting drug similarity measures exhibit varying levels of incompleteness. This incompleteness of similarity measures, which has been for the most part overlooked by prior work, is already an issue even when a single data source such as DrugBank is used. The reason is that not all the attributes needed by a given similarity measure are available for all drugs. Without any additional machine learning features, the learning algorithm cannot distinguish between a low similarity value between two drugs due to incomplete data about at least one of the drugs or real dissimilarity between them.

Problem 4: Usage of inappropriate evaluation metrics. Existing work [2, 6, 7] use mainly the area under the R.O.C curves (AUROC) as the evaluation metric to assess the quality of predictions. They often justify their decision to rely on a balanced testing dataset because of the observation that AUROC is not sensitive to the ratio of positive to negative examples. However, as shown in [9] and reinforced in our experimental evaluation section, AUROC is not appropriate for skewed distribution. Metrics designed specifically for skewed distribution such as precision & recall, F-score, or area under Precision-Recall curve (AUPR) should be used instead. Unfortunately, when prior work use these metrics, they do so on a balanced testing data set, which results in artificially high values. For example, for a trivial classifier that report all pairs of drugs as interacting, recall is 1, precision 0.5 and F-score 0.67. As shown in our evaluation, on unbalanced testing dataset (with prevalence of drug-drug interacting ranging from 10% to 30%), the basic similarity-based prediction produces excellent AUROC values, but mediocre F-score or AUPR.

To address these shortcomings, we present an extension of our system Tiresias, a large-scale similarity-based framework that predicts DDIs through link prediction [10, 11]. Tiresias begins by a semantic integration of the input data that results in a knowledge graph describing drug attributes and relationships with various related entities such as enzymes, chemical structures, and pathways. The knowledge graph is then used to compute several similarity measures between all the drugs in a scalable and distributed framework. In Tiresias, we primarily relied on a carefully engineered set of local drug similarity features derived from the information directly associated to each drug (i.e., one hop away). In this paper, we go beyond our original Tiresias by (*i*) introducing an enriched set of similarity features through extending the set of local features. The new added features include 8 new chemical structure based similarity measures, drug side effects, physiological effects, targets, metabolizing enzyme and MeSH-based similarity. More importantly, (*ii*) we enrich our knowledge graph with both structured and unstructured data sources (e.g., DailyMed[1]). We also introduce the notion of global features to capture the structural and textual features of our knowledge graph. These features are learnt by minimizing a global loss function that considers the complete structure of the knowledge graph. (*iii*) Finally, we provide a richer set of experiments to evaluate Tiresias in both newly developed and existing drugs scenarios that demonstrates the effectiveness of our newly developed local and global features. Below we summarize the key contributions of Tiresias.

Broader set of data sources: Tiresias introduces a first of kind semantic integration of a comprehensive set of structured and unstructured data sources. We exploit information originating from multiple linked data sources including, e.g., DrugBank, UMLS, DailyMed, Uniprot and CTD (cf. Section 4) to construct a knowledge graph. This integrated knowledge graph describes drug attributes and relationships with various related entities such as enzymes, chemical structures, and pathways.

Extensive set of novel similarity measures: We utilize the integrated knowledge graph to compute several similarity measures between all the drugs (cf. Section 5). We develop new drug-drug similarity measures based on various properties of drugs including metabolic and signaling pathways, drug mechanism of action and physiological effects. We also define a new class of global drug features by learning low-dimensional embeddings of drugs from textual and graph-based datasets.

Handling Data Skewness and Incompleteness: We build a large-scale and distributed linear regression learning model (in Apache Spark) to predict the existence of DDIs. Our model efficiently handle skewed distribution of DDIs and data incompleteness through ; (*i*) a combination of case control sampling for rare events and (*ii*) a new class of calibration features. First, in Section 6, we present a systematic methodology to estimate that the true prevalence of interacting drug pairs in the set of all drug pairs, which we discover to be ranging between 10%

[1] `https://dailymed.nlm.nih.gov/dailymed/`

and 30%. Second, to address the incompleteness of similarity measures which affects prediction quality as measured by precision & recall, F-score, etc, we introduce a new class of features, called calibration features (cf. Section 5.3) that captures the relative completeness of the drug-drug similarity measures.

Extending prediction to newly developed drugs: Given Tiresias extensive set of similarity-based features, we demonstrate that our framework is capable of dealing with drugs without any known interacting drugs. We further show that techniques developed in Tiresias significantly improve the prediction quality for new drugs not seen at training.

Comprehensive evaluation: We conduct detailed evaluations with real data assuming skewed data distribution and using proper evaluation metrics including precision, recall, F-score and AUPR. For newly developed drugs, using standard 10-fold cross validation, Tiresias is able to achieve DDI prediction with an average F-Score of 0.74 (vs. 0.65 for the baseline) and area under PR curve of 0.82 (vs. 0.78 for the baseline). For the existing drugs scenario, Tiresias is able to achieve an F-Score value of 0.85 (vs. 0.75 for the baseline) and AUPR of 0.92 (vs. 0.87 for the baseline). The performance becomes even better as we include our global embedding-based features; F-score increases to 0.89 while AUPR increases to 0.97. Additionally, we introduce a novel retrospective analysis to demonstrate the effectiveness of our approach to predict correct, but yet unknown DDIs. Up to 68% of all DDIs found after 2011 were correctly predicted using only DDIs known in 2011 as positive examples in training (cf. Section 7).

The rest of the paper is organized as follows. Section 2 discusses the preliminaries of similarity-based DDI approaches as well as text and graph embedding techniques. In Section 3, we give an overview of the main components of Tiresias and highlight its computation phases. Section 4 describes the data integration phase and how Tiresias handles the associated integration challenges. Then, we describe the different extracted features required for model building in Section 5. Section 6 shows how Tiresias handles unbalanced data distributions while Section 7 presents the experimental evaluation. Finally, Section 8 surveys the related work and we conclude in Section 9.

2. Background

In this section, we discuss the main ideas of similarity-based DDI approaches. We also discuss a recent line of research which aims at learning low-dimensional embeddings of entities of textual and graph-based datasets. These embedding approaches are used in Tiresias to define a new family of global features for comparing drugs.

2.1. Similarity-based DDI predictions

Similar to content-based recommender systems, the core idea of similarity-based approaches [2, 6, 7] is to predict the existence of an interaction between a candidate pair of drugs by comparing it against known interacting pairs of drugs. Finding known interacting drugs that are very similar to the candidate pair provides supporting evidence in favor of the existence of a drug-drug interaction between the two candidate drugs.

These approaches first define a variety of drug similarity measures to compare drugs. A drug similarity measure sim is a function that takes as input two drugs and returns a real number between 0 (no similarity between the two drugs) and 1 (perfect match between the two drugs) indicating the similarity between the two drugs. SIM denotes the set of all drug similarity measures. Entities of interest for drug-drug interaction prediction are not single drugs, but rather pair of drugs. Thus, drug similarity measures in SIM need to be extended to produce drug-drug similarity measures that compare two pairs of drugs (e.g., a pair of candidate drugs against an already known interacting pair of drugs). Given two drug similarity measures sim_1 and sim_2 in SIM, we can define a new drug-drug similarity measure, denoted $sim_1 \otimes sim_2$, that takes as input a two pairs of drugs (a_1, a_2) and (b_1, b_2) and returns the similarity between the two pairs of drugs computed as follows:

$$sim_1 \otimes sim_2((a_1, a_2), (b_1, b_2)) = avg(sim_1(a_1, b_1), sim_2(a_2, b_2))$$

where avg is an average or mean function such as the geometric mean or the harmonic mean. In other words, the first drug similarity measure (sim_1) is used to compare the the first element of each pair and the second drug similarity measure (sim_2) is used to compare the second element of each pair. Finally, the results of the two comparisons are combined using, for example, harmonic or geometric mean. The set of all drug-drug similarity measures thus defined by combining drug similarity measures in SIM is denoted $SIM^2 = \{sim_1 \otimes sim_2 | sim_1 \in SIM \wedge sim_2 \in SIM\}$.

Given a set $KDDI$ of known drug-drug interactions, a drug-drug similarity measure $sim_1 \otimes sim_2 \in SIM^2$, and a candidate drug pair (d_1, d_2), the prediction based solely on $sim_1 \otimes sim_2$ that d_1 and d_2 interacts, denoted $predict[sim_1 \otimes sim_2, KDDI](d_1, d_2)$, is computed as the arithmetic mean of the similarity values between (d_1, d_2) and the top-k most similar known interacting drug pairs to (d_1, d_2): $predict[sim_1 \otimes sim_2, KDDI](d_1, d_2) =$

$$amean(top_k\{sim_1 \otimes sim_2((d_1, d_2), (x, y)) | (x, y) \in KDDI - \{(d_1, d_2)\}\})$$

where $amean$ is the arithmetic mean, and, in most cases, k is equal to 1. The power of similarity-based approaches stems from not relying on a single similarity based prediction, but from combining all the individual independent predictions $predict[sim_1 \otimes sim_2, KDDI]$ for all $sim_1 \otimes sim_2 \in KDDI$ into a single score that indicates the level of confidence in the existence of a drug-drug interaction. This combination is typically done through machine learning (e.g., logistic regression): the training is performed using $KDDI$ as the ground truth and, given a drug pair (d_1, d_2), its feature vector consists of $predict[sim_1 \otimes sim_2, KDDI](d_1, d_2)$ for all $sim_1 \otimes sim_2 \in KDDI$.

Similarity-based methods have a number of clear advantages: (*i*) compared with direct feature vector-based approaches, similarity-based approaches do not need complex and difficult feature extraction or selection (e.g., generating and combining features for a drug pair); (*ii*) many drug similarity measures such as chemical structure similarity [4], target protein similarity [5], and side-effect similarity [6] have already been fully developed and are widely used; (*iii*) similarity-based approaches can be directly related to well-developed kernel methods, which can provide high-performance prediction results; (*iv*) different similarity measures can easily be combined. For example, we generate drug chemical-protein interactome (CPI) similarity measure based on the concept of DDI-CPI server [3], which shows the flexibility of our method in integrating multiple drug information resources.

2.2. Textual Embedding

Recently, the word2vec model has attracted a lot of attention to construct embedding for textual data [12]. It aims at learning high-quality word vectors (embedding) from huge data sets with billions of words. Word2vec is a two-layer neural network used for computing vector representations of words. Each word vector is trained to maximize the log probability of the word given the context word(s) occurring within a fixed-size window. Word2vec proposed two different architectures that can be utilized to obtain word vectors; Continuous Bag-of-Words (CBOW) and Skip-gram Model [12]. CBOW tries to predict the word given its context while skip-gram tries to predict the context given a word. In skip-gram, the context is not limited to the word's immediate context, rather training instances can be created by skipping a constant number of words. CBOW is several times faster compared to skip-gram which tends to be more accurate [12] due to the more generalizable contexts generated.

The training objective of the Skip-gram model is to find word representations that are useful for predicting the surrounding words in a sentence or a document. Given a sequence of words $w_1, w_2, w_3, \ldots, w_T$, the Skip-gram model tries to to maximize the average log probability as follows:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

where c is the size of the training context while $p(w_{t+j}|w_t)$ is defined using the softmax function as follows:

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^{\top} v_{w_I}\right)}$$

v_w and v'_w refer to the input and output vector representations of w, respectively while W is the number of words in the vocabulary. v_w comes from input → hidden layer weight matrix while v'_w comes from hidden → output layer weight matrix. The training objective is to maximize the conditional probability of observing the actual output word w_O given the input context word w_I with regard to the weights. With the existence of two vector representations for each word in the vocabulary, learning the input vectors becomes cheap; but learning the output vectors is very expensive. Consequently, more efficient approximation techniques like hierarchical softmax and negative sampling [12] can be utilized.

One of the advantages of word2vec is its ability to groups similar words together in the vector space. Moreover, it can automatically learn concepts and predict semantic relationships between words using simple algebraic operations on the word vectors [12, 13]. For example, vector(Germany) + vector(capital) is close to vector(Berlin) and vector(Russia) + vector(river) is close to vector(Volga River). Furthermore, vector(King) - vector(Man) + vector(Woman) results in a vector close to vector(Queen). Similarly, vector(Einstein) - vector(scientist) + vector(Picasso) results in a vector close to vector(painter).

2.3. Graph Embedding

Several approaches [14, 15, 16] have been proposed to embed an input knowledge graph into a continuous vector space while preserving certain properties of the original graph. The output of these techniques is a vector representation for each entity and relation in the input graph. Each entity is represented as a point in the vector space while each relation is modeled as an operation (e.g. translation, projection, etc) in that space.

TransE [14] is an energy-based model for learning low-dimensional embeddings of entities. TransE treats a triple $(\mathsf{s}, \mathsf{p}, \mathsf{o})$ as a relation-specific translation from a head entity (subject) to a tail entity (object). The translation function is modeled as a simple addition of the vectors that correspond to the head entity (s) and the predicate relation (p). When $(\mathsf{s}, \mathsf{p}, \mathsf{o})$ holds, TransE tries to have o as the nearest neighbor of $\mathsf{s} + \mathsf{p}$ and far away otherwise. To learn these embedding, TransE minimizes the max-margin-based ranking cost function as follows:

$$\mathcal{L} = \sum_{(\mathsf{s},\mathsf{p},\mathsf{o})\in S}\ \sum_{(\mathsf{s}',\mathsf{p},\mathsf{o}')\in S'_{(\mathsf{s},\mathsf{p},\mathsf{o})}} [\gamma + d(\mathsf{s} + \mathsf{p}, \mathsf{o}) - d(\mathsf{s}' + \mathsf{p}, \mathsf{o}')]_+$$

where $\mathcal{L}$ is the loss function to be minimized, $\gamma > 0$ is a margin hyperparameter and d finds similarity of the translation and the object embedding which can be measured by the L_1 or L_2 distance. $S'_{(\mathsf{s},\mathsf{p},\mathsf{o})}$ is the set of corrupted triples which is drawn from the set of training triplets with either the head or tail replaced by a random entity (but not both at the same time). TransE is an easy to train model since it relies on a reduced set of parameters. At the same time, it is an efficient model that achieves high prediction performance. TransH [15] improves the performance of TransE when dealing with relations with mapping properties of reflexive/one-to-many/many-to-one/many-to-many. TransH shows better performance compared to TransE at the cost of a higher computational complexity.

Recently, Nickel et al. introduced HolE [16], a compositional vector space based model for knowledge graphs. The meaning and representation of entities in compositional models do not vary according to their position in the compositional representation. Furthermore, the representations of all entities and relations are learned jointly. This allows the model to propagate information between triples which captures global dependencies in the data. HolE combines the expressive power of the tensor product with the efficiency and simplicity of TransE. It represents a pair of entities (a, b) using the circular correlation (a compression of the tensor product) of their vectors as follows:

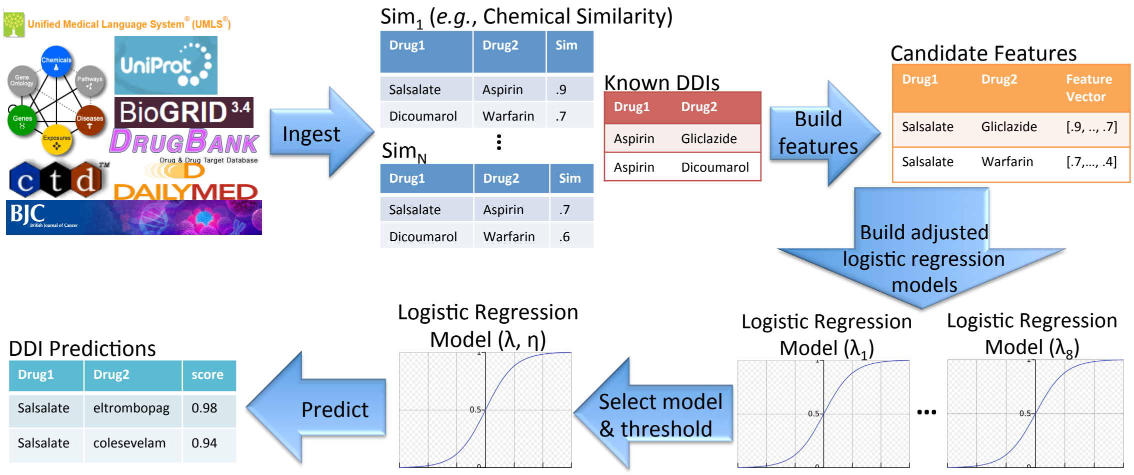

Unified Medical Language System® (UMLS®)
UniProt
BioGRID 3.4
DrugBank
ctd
DailyMed
BJC
Ingest
Sim1 (e.g., Chemical Similarity)
Drug1 Drug2 Sim
Salsalate Aspirin .9
Dicoumarol Warfarin .7
SimN
Drug1 Drug2 Sim
Salsalate Aspirin .7
Dicoumarol Warfarin .6
Known DDIs
Drug1 Drug2
Aspirin Gliclazide
Aspirin Dicoumarol
Build features
Candidate Features
Drug1 Drug2 Feature Vector
Salsalate Gliclazide [.9, .., .7]
Salsalate Warfarin [.7,..., .4]
Build adjusted logistic regression models
Logistic Regression Model (λ1)
Logistic Regression Model (λ8)
Select model & threshold
Logistic Regression Model (λ, η)
Predict
DDI Predictions
Drug1 Drug2 score
Salsalate eltrombopag 0.98
Salsalate colesevelam 0.94

Figure 1: Overview of Tiresias: a large-scale similarity-based DDI prediction framework.

$$\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{a} \star \boldsymbol{b},$$

where $\star : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ denotes circular correlation which is defined as:

$$[\boldsymbol{a} \star \boldsymbol{b}]_k = \sum_{i=0}^{d-1} a_i b_{(k+i) \bmod d}.$$

Consequently, the probability of a triple is modeled as:

$$\Pr(\phi_p(\mathsf{s}, \mathsf{o}) = 1 | \Theta) = \sigma(\boldsymbol{r}_p^\top (\boldsymbol{e}_s \star \boldsymbol{e}_o)).$$

where $\boldsymbol{r}_p$ and $\boldsymbol{e}_i$ are vector representations of relations and entities. $\sigma(x) = 1/(1+\exp(-x))$ denotes the logistic function while $\Theta = \{\boldsymbol{e}_i\}_{i=1}^{n_e} \cup \{\boldsymbol{r}_k\}_{k=1}^{n_r}$ denotes the set of all embeddings. $\star$ denotes the compositional operator which creates a composite vector representation for the pair (s, o) from the embeddings $\boldsymbol{e}_s, \boldsymbol{e}_o$. HolE is shown to handle relatively large knowledge graphs and provide better performance compared to state-of-the-art embedding techniques.

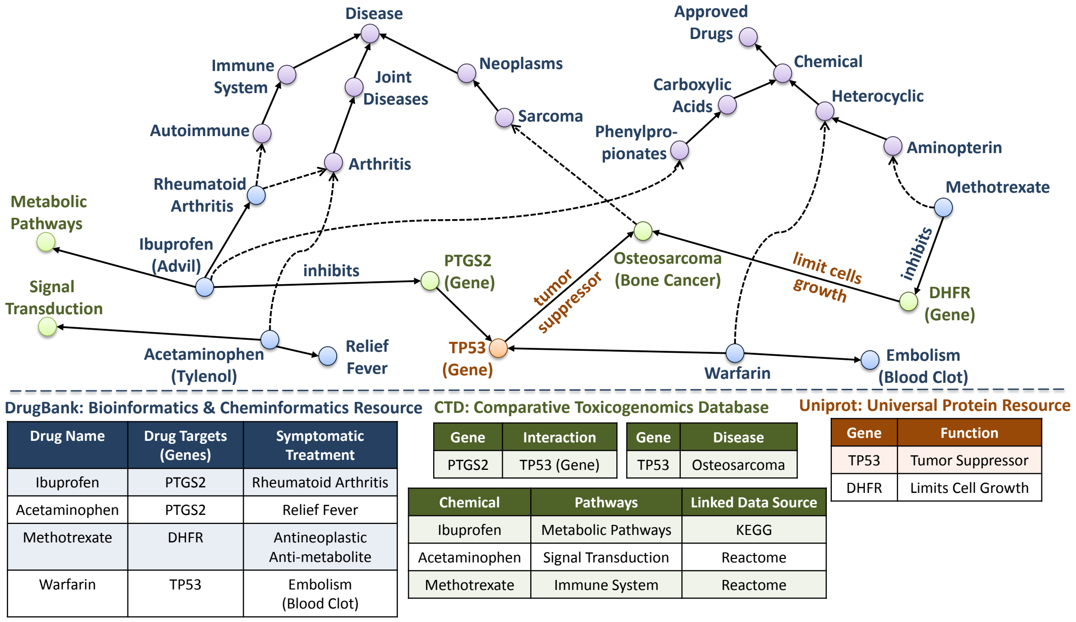
3. Tiresias Overview

Figure 1 shows the architecture of Tiresias. It consists of five key phases (the arrows in Figure 1). We describe below each phase in details.

Ingestion: In this phase, data originating from multiple sources are ingested and integrated to create various drug similarity measures (represented as blue tables in Figure 1) and a known DDIs table. Similarity measures are not necessarily complete in the sense that some drug pairs may be missing from the similarity tables displayed in Figure 1. The known DDIs table, denoted *KDDI*, contains the set of 12,104 drug pairs already known to interact in DrugBank. In the 10 fold cross validation of our approach, *KDDI* is randomly split into 3 disjoint subsets: $KDDI_{train}$, $KDDI_{val}$, and $KDDI_{test}$ representing the set of positive examples respectively used in the training, validation and testing (or prediction) phases. Contrary to most prior work, which partition *KDDI* on the DDI associations instead of on drugs, our partitioning simulates the scenario of the introduction of newly developed drugs for which no interacting drugs are known. In particular, each pair (d_1, d_2) in $KDDI_{test}$ is such that either d_1 or d_2 does not appear in $KDDI_{train}$ or KDD_{val}.

Feature Building: Given a pair of drugs (d_1, d_2), we construct its machine learning feature vector derived from the drug similarity measures and the set of DDIs known at training. Like previous similarity-based approaches, for a drug candidate pair (d_1, d_2) and a drug-drug similarity measure $sim_1 \otimes sim_2 \in SIM^2$, we create a feature that indicates the similarity value of the known pair of interacting drugs most similar to (d_1, d_2) (see Section 5.2). Unlike prior work, we introduce new calibration features to address the issue of the incompleteness of the similarity measures and to provide more information about the distribution of the similarity values between a drug candidate pair and all known interacting drug pairs - not just the maximum value (see Section 5.3).

Logistic Regression Model: As a result of relying on more data sources, using more similarity measures, and introducing new calibration features, we have significantly more features (1014) than prior work (e.g., [2] uses only 49 features). Thus, there is an increased risk of overfitting that we address by performing L_2-model regularization. Since the optimal regulariza-



DrugBank: Bioinformatics & Cheminformatics Resource

Drug Name	Drug Targets (Genes)	Symptomatic Treatment
Ibuprofen	PTGS2	Rheumatoid Arthritis
Acetaminophen	PTGS2	Relief Fever
Methotrexate	DHFR	Antineoplastic Anti-metabolite
Warfarin	TP53	Embolism (Blood Clot)

CTD: Comparative Toxicogenomics Database

Gene	Interaction
PTGS2	TP53 (Gene)

Gene	Disease
TP53	Osteosarcoma

Chemical	Pathways	Linked Data Source
Ibuprofen	Metabolic Pathways	KEGG
Acetaminophen	Signal Transduction	Reactome
Methotrexate	Immune System	Reactome

Uniprot: Universal Protein Resource

Gene	Function
TP53	Tumor Suppressor
DHFR	Limits Cell Growth

Figure 2: Semantic curation and linkage of data from variety of sources on the Web.

tion parameter is not known a-priori, in the model generation phase, we build 8 different logistic regression models using 8 different regularization values. To address issues related to the skewed distribution of DDIs (for an assumed prevalence DDIs lower than 17%), we make some adjustments to logistic regression (see Section 6).

Model Selection: The goals of this phase are twofold. First, in this phase, we select the best of the eight models (i.e., the best regularization parameter value) built in the model generation phase by choosing the model producing the best F-score on the validation data. Second, we also select the optimal threshold as the threshold at which the best F-score is obtained on the validation data evaluated on the selected model.

Prediction: Let f denote the logistic function selected in the model validation phase and η the confidence threshold selected in the same phase. In the prediction phase, for each candidate drug pair (d_1, d_2), we first get its feature vector v computed in the feature construction phase. $f(v)$ then indicates the probability that the two drugs d_1 and d_2 interact, and the pair (d_1, d_2) is labeled as interacting iff. $f(v) \geq \eta$.

4. Data Integration

4.1. Datasets

We form our knowledge graph by integrating data from a variety of web sources together. These sources come in different formats including XML, relational, graph and CSV formats . As partially shown in Figure 2, our data comes from variety of sources: (*i*) *DrugBank* [17]: it offers data about known drugs and diseases. (*ii*) *DailyMed*[2] provides high qualitative information about marketed drugs in the United States. (*iii*) *Comparative Toxicogenomics Database* [18] provides information about gene interaction. (*iv*) *Uniprot* [19] provides details about the functions and structure of genes. (*v*) *BioGRID* database collects genetic and protein interactions [20]. (*vi*) *Unified Medical Language System* [21] is the largest repository of biomedical vocabularies including *NCBI* taxonomy, *Gene Ontology (GO)*. (*vii*) *Medical Subject Headings (MeSH)* [22], and (*viii*) *National Drug File - Reference Terminology (NDF-RT)* classifies drug with a multi-category reference models such as cellular or molecular interactions and therapeutic categories [23].

4.2. Addressing Integration Challenges

One of the salient feature of our Tiresias framework is to leverage many available sources on the Web. More importantly, there is a crucial need to connect these disparate sources in order to create a knowledge graph that is continuously being enriched as ingesting more sources. Notably the life science community has already recognized the importance of the data integration and taken the first step to employ the Linked Open Data methodology for connecting identical entities across different sources. However, most of the existing linkages in the scientific domain are often done statically, which results in many outdated or even non-existent links overtime.

Therefore, even when the data is presumably linked, we are forced to verify these links. Furthermore, there are number of fundamental challenges that must be addressed to construct a unified view of the data with rich interconnectedness and semantics [24] — a knowledge graph. For example, we employ *entity resolution* methodology either through syntactical disambiguation (e.g., cosine similarity, edit distance, or language model techniques [25]) or through semantic analysis by examining the conceptual property of entities [21]. These techniques are not only essential to identify similar entities but also instrumental in designing and capturing similarities among entities in order to engineer features necessary to enable DDIs prediction.

[2] `https://dailymed.nlm.nih.gov/dailymed/`

As part of our knowledge graph curation task, we identify which attributes or columns refer to which real world entities (i.e., data instances). Therefore, our constructed knowledge graph possess a clear notion of what the entities are, and what relations exist for each instance in order to capture the data interconnectedness. These may be relations to other entities, or the relations of the attributes of the entity to data values. As an example, in our ingested and curated data, we have a table for *Drug*, and have the columns *Name*, *Targets*, *Symptomatic Treatment*. Our knowledge graph has an identifier for a real world drug *Methotrexate*, and captures its attributes such as *Molecular Structure* or *Mechanism of Actions*, as well as relations to other entities including *Genes* that *Methotrexate* targets (e.g., *DHFR*), and subsequently, *Conditions* that it treats such as *Osteosarcoma (bone cancer)* that are reachable through its target genes, as demonstrated in Figure 2. We then encode and store the integrated graph in RDF format which is used as input to Apache Spark for similarity calculation and model building. Constructing a rich knowledge graph is a necessary step before building our predication model as discussed next.

5. Feature Engineering

In this section, we describe the drug similarity measures used to compare drugs and how various machine learning features are generated from them.

5.1. Drug Similarity and Drug-Drug Similarity Measures

To measure the similarity between two drugs, Tiresias uses a set of features that are divided into two categories based on the way they are generated; local and global similarity-based features. Local features are the set of features engineered based on the information available about drugs. These features consider the direct associated information with each drug; e.g. chemical structure, side effects and drug target. On the other hand, global features are obtained by embedding drugs in low-dimensional vector spaces. We learn a vector representation for each drug such that the similarity between two drugs is defined as the cosine similarity between their corresponding vectors. To construct these vector representations, we minimize a global loss function that considers all facts (including structural properties of graphs) in the dataset. We describe below each category in details.

5.1.1. Local Similarity-based Features

Based on the available information about drugs, we manually selected the following drug similarity measures to compare two drugs.

Chemical-Protein Interactome (CPI) Profile based Similarity: The Chemical-Protein Interactome (CPI) profile of a drug d, denoted $cpi(d)$, is a vector indicating how well its chemical structure docks or binds with about 611 human Protein Data Bank (PDB) structures associated with drug-drug interactions [3]. The CPI profile based similarity of two drugs d_1 and d_2 is computed as the cosine similarity between the mean-centered versions of vectors $cpi(d_1)$ and $cpi(d_2)$.

Mechanism of Action based Similarity: For a drug d, we collect all its mechanisms of action obtained from NDF-RT. To discount popular terms, Inverse Document Frequency (IDF) is used to assign more weight to relatively rare mechanism of actions: $IDF(t, Drugs) = log\frac{|Drugs|+1}{DF(t,Drugs)+1}$ where $Drugs$ is the set of all drugs, t is a mechanism of action, and $DF(t, Drugs)$ is the number of drugs with the mechanism of action t. The IDF-weighted mechanism of action vector of a drug d is a vector $moa(d)$ whose components are mechanisms of action. The value of a component t of $moa(d)$, denoted $moa(d)[t]$, is zero if t is not a known mechanism of action of d; otherwise, it is $IDF(t, Drugs)$. The mechanism of action based similarity measure of two drugs d_1 and d_2 is the cosine similarity of the vectors $moa(d_1)$ and $moa(d_2)$.

Physiological Effect based Similarity: For a drug d, we collect all its physiological effects obtained from NDF-RT. The physiological effect based similarity measure of two drugs d_1 and d_2 is defined as the cosine similarity of IDF-weighted physiological effect vectors of the two drugs - which are computed in the same way as the IDF-weighted mechanism of action vector described in the previous paragraph.

Pathways based Similarity: Information about pathways affected by drugs is obtained from CTD database. The pathways based similarity of two drugs is defined as the cosine similarity between the IDF-weighted pathways vectors of the two drugs, which are computed in a similar way as IDF-weighted mechanism of action vectors.

Side Effect based Similarity: Side effects associated with a drug are obtained from SIDER database of drug side effects [26]. The side effect based similarity of two drugs is defined as the cosine similarity between the IDF-weighted side effect vectors of the two drugs, which are computed in a similar way as IDF-weighted mechanism of action vectors of drugs.

Metabolizing Enzyme based Similarities: Information about enzymes responsible for the metabolism of drugs is obtained from DrugBank. We define two drug similarity measures related to metabolizing enzymes.

- The first measure compares drugs based on the commonality of the metabolizing enzymes they interact with. However, it does not take into account the nature of the interaction (i.e., inhibitor, substrate, or inducer). It is formally defined as the cosine similarity between the IDF-weighted metabolizing enzyme vectors of two drugs, which are computed in a similar way as the IDF-weighted mechanism of action vectors of drugs.

- The second measure takes into account the nature of the interaction. For example, if drug d_1 interacts with a single metabolizing enzyme e by acting as an inhibitor, and drug d_2 also interacts only with the same enzyme e but as an inducer. According to the first measure, d_1 and d_2 will have a similarity value of 1. However, once the nature of the interaction with the enzyme is taken into account, it is clear that d_1 and d_2 are actually very dissimilar. Formally, to take into account the nature of the interaction, we modify the IDF-weighted metabolizing enzyme vector $me(d)$ of a drug d by multiplying by -1 the value of each component corresponding to an enzyme that is inhibited by the drug. The similarity between two drugs is then defined as the normalized cosine similarity between the modified IDF-weighted metabolizing enzyme vectors of the two drugs (normalization ensures that the value remains in the [0, 1] range instead of [-1, 1] range).

Drug Target based Similarities: Information about proteins targeted by a drug is obtained from DrugBank. We define three drug similarity measures related to drug targets. The first two are constructed in a similar way as the two metabolizing enzyme based similarities. The first similarity ignores the nature of the action of the drug on an protein target (i.e., inhibition or activation), whereas the second takes it into account. The third similarity measure compares drugs based on the molecular functions of their protein targets as defined in Uniprot using Gene Ontology (GO) annotations. Specifically, the third similarity measure is computed as Resnik semantic similarity [27], using the csbl.go R package [28].

Chemical Structure Similarity: Fingerprinting is considered nowadays an important tool for judging the similarity of drugs chemical structures. Therefore, we define a new similarity measure for comparing two drugs based on the fingerprints of their chemical structures. The chemical structures of the drugs are obtained from DrugBank in the SMILES format. We use the Chemical Development Kit[3] (CDK) [29], with default setting, to compute the fingerprints of the molecular structures of drugs as bit vectors. Then, the chemical structure similarity of two drugs is computed as the Jaccard similarity (or Tanimoto coefficient) of their fingerprints. There are several approaches for computing the fingerprints. Instead of using a single method, we use 9 types of fingerprints: path-based, circular, shortest path, MACCS, EState, Extended, KlekotaRoth, Pubchem and substructure Fingerprinter. More details about each fingerprint type can be found in [30].

Anatomical Therapeutic Chemical (ATC) Classification System based Similarity: ATC [31] is a classification of the active ingredients of drugs according to the organs that they affect as well as their chemical, pharmacological and therapeutic characteristics. The classification consists of multiple trees representing different organs or systems affected by drugs, and different therapeutical and chemical properties of drugs. The ATC codes associated with each drug are obtained from DrugBank. For a given drug, we collect all its ATC code from DrugBank to build a ATC code vector (the most specific ATC codes associated with the drug -i.e., leaves of the classification tree- and also all the ancestor codes are included). The ATC based similarity of two drugs is defined as the cosine similarity between the IDF-weighted ATC code vectors of the two drugs, which are computed in a similar way as IDF-weighted mechanism of action vectors.

MeSH based Similarity: DrugBank associates each drug with a set of relevant MeSH [22] (Medical Subject Heading) terms. The MeSH based similarity of two drugs is defined as the cosine similarity between the IDF-weighted MeSH vectors of the two drugs, which are computed in a similar way as IDF-weighted mechanism of action vectors of drugs.

5.1.2. Global Similarity-based Features

Tiresias relies on another set of features that are used to compare two drugs. We used graph and word embedding techniques (see Section 2) to get a vector representation for each drug. Recall that these techniques minimize a global loss function that consider all the facts in the dataset. Therefore, the obtained vector representation for the entities capture global dependencies in the data and inherit semantics that goes beyond just considering the direct neighbours. We describe below how we utilize these techniques to define new set of global similarity features for comparing drugs.

Word Embedding-based Features: We exploit DailyMed[4] and DrugBank[5] to construct textual embedding for each drug in order to obtain a numerical representation in the vector space model; thus, enabling a computational framework to compare drugs in the learned embedding space.

To avoid any possible information leakage, we remove the drug interaction information from each drug in each dataset. Then, we use word2vec [12] on both DailyMed and DrugBank corpora to obtain a vector representation for each drug. We use the Skip-gram architecture since it is proved to show a more accurate representation compared to CBOW. As a result, we have a different vector representation for each drug; one per each database. Then, we define a word-embedding based similarity between a pair of drugs (d_1, d_2) which is calculated as the cosine similarity between the the two vectors that correspond to d_1 and d_2, respectively.

Graph Embedding-based Features: Similarly, we utilize graph embedding techniques to learn a vector representation of the drugs from our integrated knowledge graphs (see Figure 2). We use two graph embedding techniques; TransH [15] and HolE [16]. For each method, we define a drug-drug similarity measure that is calculated as the cosine similarity between the corresponding drugs vectors. We show in Section 7 the effect of adding these features to Tiresias.

Most of the previously defined drug similarity measures rely on both cosine similarity and IDF (to discount popular terms). We have evaluated our system by replacing cosine by other similarity metrics such as weighted Jaccard or soft cosine similarity [32] (when components of the vectors are elements of a taxonomical hierarchy: e.g., Mechanism of Action or Physiological Effect) without any noticeable improvement of the quality of our predictions. We have also tried using information theoretical means to discount popular terms (e.g., entropy based weighting) instead of IDF without any noticeable improvement of the quality of our predictions.

[3] http://cdk.github.io/cdk/

[4] https://dailymed.nlm.nih.gov/dailymed/

[5] http://www.drugbank.ca/releases/latest

The set of all drug similarity measures is denoted SIM. As explained in the background section 2, drug similarity measures in SIM need to be extended to produce drug-drug similarity measures that compare two pairs of drugs (e.g., a pair of candidate drugs against an already known interacting pair of drugs). SIM^2 denotes the set of all drug-drug similarity measures derived from SIM as explained in section 2.

5.2. Top-k Similarity-based Features

Like previous similarity-based approaches, for a given drug candidate pair (d_1, d_2), a set KDD_{train} of DDIs known at training, and a drug-drug similarity measure $sim_1 \otimes sim_2 \in SIM^2$, we create a similarity-based feature, denoted $abs_{sim_1 \otimes sim_2}$ and computed as the similarity value between (d_1, d_2) and the most similar known interacting drug pair to (d_1, d_2) in $KDDI_{train}$. In other words,

$$abs_{sim_1 \otimes sim_2}(d_1, d_2) = max(D_{sim_1 \otimes sim_2}(d_1, d_2))$$

where $D_{sim_1 \otimes sim_2}(d_1, d_2)$ is the set of all the similarity values between (d_1, d_2) and all known DDIs:

$$\begin{aligned} D_{sim_1 \otimes sim_2}(d_1, d_2) = \{&sim_1 \otimes sim_2((d_1, d_2), (x, y)) \\ &\mid (x, y) \in KDDI_{train} - \{(d_1, d_2)\}\} \end{aligned} \tag{1}$$

Note that these similarity-based features are computed using only DDIs known at training (i.e., $KDDI_{train}$)

5.3. Calibration Features

Calibration of top-k similarity-based features: For a drug candidate pair (d_1, d_2), a high value of the similarity-based feature $abs_{sim_1 \otimes sim_2}(d_1, d_2)$ is a clear indication of the presence of at least one known interacting drug pair very similar to (d_1, d_2) according to the drug-drug similarity measure $sim_1 \otimes sim_2$. However, this feature value provides to the machine learning algorithm only a limited view of the distribution $D_{sim_1 \otimes sim_2}(d_1, d_2)$ of all the similarity values between (d_1, d_2) and all known DDIs (see equation (1)).

For example, with only access to $max(D_{sim_1 \otimes sim_2}(d_1, d_2))$, there is no way to differentiate between a case where that maximum value is a significant outlier (i.e., many standard deviation away from the mean of $D_{sim_1 \otimes sim_2}(d_1, d_2)$) and the case where it is not too far from the mean value of $D_{sim_1 \otimes sim_2}(d_1, d_2)$. Since it would be impractical to have a feature for each data point in D (overfitting and scalability issues), we instead summarize the distribution $D_{sim_1 \otimes sim_2}(d_1, d_2)$ by introducing the following features to capture its mean and standard deviation:

$$avg_{sim_1 \otimes sim_2}(d_1, d_2) = mean(D_{sim_1 \otimes sim_2}(d_1, d_2))$$

$$std_{sim_1 \otimes sim_2}(d_1, d_2) = stdev(D_{sim_1 \otimes sim_2}(d_1, d_2)))$$

To calibrate the absolute maximum value computed by $abs_{sim_1 \otimes sim_2}(d_1, d_2)$, we introduce a calibration feature, denoted $rel_{sim_1 \otimes sim_2}$, that corresponds to the z-score of the maximum similarity value of the candidate and a known DDI (i.e., it indicates the number of standard deviations $max(D)$ is from the mean of D):

$$rel_{sim_1 \otimes sim_2}(d_1, d_2) = \frac{abs_{sim_1 \otimes sim_2}(d_1, d_2) - avg_{sim_1 \otimes sim_2}(d_1, d_2)}{std_{sim_1 \otimes sim_2}(d_1, d_2)}$$

Finally, for a candidate pair (d_1, d_2), we add a boolean feature, denoted $con_{sim_1 \otimes sim_2}(d_1, d_2)$, that indicates whether the most similar known interacting drug pair contains d_1 or d_2.

Calibration of drug-drug similarity measures: Features described so far capture similarity values between a drug candidate pair and known DDIs. As such, a high feature value for a given candidate pair (d_1, d_2) does not necessarily indicate that the two drugs are likely to interact. For example, it could be the case that, for a given drug-drug similarity measure, (d_1, d_2) is actually very similar to most drug pairs (whether or not they are known to interact). Likewise, a low feature value does not necessarily indicate a reduced likelihood of drug-drug interaction if (d_1, d_2) has a very low similarity value with respect to most drug pairs (whether or not they are known to interact). In particular, such a low overall similarity between (d_1, d_2) and most drug pairs is often due to the incompleteness of the similarity measures considered. For a drug-drug similarity measure $sim_1 \otimes sim_2 \in SIM^2$ and a candidate pair (d_1, d_2), we introduce a new calibration feature, denoted $base_{sim_1 \otimes sim_2}$, to serve as a baseline measurement of the average similarity measure between the candidate pair (d_1, d_2) and any other pair of drugs (whether or not they are known to interact). The exact expression of $base_{sim_1 \otimes sim_2}(d_1, d_2)$ is as follows:

$$\frac{\sum_{(x,y) \neq (d_1, d_2) \wedge x \neq y} sim_1 \otimes sim_2((d_1, d_2), (x, y))}{|Drugs|(|Drugs|-1)/2 - 1}$$

The evaluation of this expression is quadratic in the number of drugs $|Drugs|$, which results in a significant runtime performance degradation without any noticeable gain in the quality of the predictions as compared to the following approximation of $base_{sim_1 \otimes sim_2}$ (with a linear time complexity):

$$base_{sim_1 \otimes sim_2}(d_1, d_2) \approx hm(\frac{\sum_{x \neq d_1} sim_1(d_1, x)}{|Drugs|-1}, \frac{\sum_{y \neq d_2} sim_2(d_2, y)}{|Drugs|-1})$$

where hm denotes the harmonic mean. In other words. $base_{sim_1 \otimes sim_2}(d_1, d_2)$ is approximated as the harmonic mean of 1) the arithmetic mean of the similarity between d_1 and all other drugs computed using sim_1, and 2) the arithmetic mean of the similarity between d_2 and all other drugs computed using sim_2.

6. Dealing with Unbalanced Data

In evaluating any machine learning system, the testing data should ideally be representative of the real data. In particular, for our binary classifier that predicts whether a pair of drugs interacts, the fraction of positive examples in the testing data should be as close as possible to the prevalence or fraction of DDIs in the set of all pairs of drugs. Although the ratio of positive to negative examples in the testing has limited impact on the area under the ROC curves, as shown in the experimental evaluation, it has significant impact on other key quality metrics more appropriate for skewed distributions (e.g., precision & recall, F-score and area under precision-recall curves).

Unfortunately, the exact prevalence of DDIs in the set of all drugs pairs is unknown. Here, we provide upper and lower bounds on the true prevalence of DDIs in the set of all drug pairs. Then, we discuss logistic regression adjustments to deal with the skewed distribution of DDIs.

Upper bound: FDA Adverse Event Reporting System (FAERS) is a database that contains information on adverse events submitted to FDA. It is designed to support FDA's post-marketing safety surveillance program for drugs and therapeutic biological products. Mined from FAERS, TWOSIDES [33] is a dataset containing only side effects caused by the combination of drugs rather than by any single drugs. Used as the set of known DDIs, TWOSIDES [33] contains many false positives as some DDIs are observed from FAERS, but without rigorous clinical validation. Thus, we use TWOSIDES to estimate the upper bound of the DDI prevalence. There are 645 drugs and 63,473 distinct pairwise DDIs in the dataset. Thus, the upper bound of the DDI prevalence is about 30%.

Lower bound: We used a DDI data set from Gottlieb et al [2] to estimate the lower bound of the DDI prevalence. The data set were extracted from DrugBank [17] and the http://drugs.com website (excluding DDIs tagged as minor), updated by *CernerMultum*^{TM}. DDIs from this data set are extracted from drug's package inserts (accurate but far from complete), thus there are some false negatives in such a data set. There are 1,227 drugs and 74,104 distinct pairwise DDIs in the dataset. Thus the lower bound of the DDI prevalence is about 10%.

Modified logistic regression to handle unbalanced data: For a given assumed low prevalence of DDIs τ_a, it is often advantageous to train our logistic regression classifier on a training set with a higher fraction τ_t of positive examples and to later adjust the model parameters accordingly. The main motivation for this *case-control sampling* approach for rare events [34] is to improve runtime performance of the model building phase since, for the same number of positive examples, the higher fraction τ_t of positive examples yields a smaller total number of examples at training. Furthermore, for an assumed prevalence $\tau_a \leq 0.17$, the quality of the predictions is only marginally affected by the use of a training set with a ratio of one positive example to 5 negative examples (i.e., $\tau_t \sim 0.17$)

A logistic regression model with parameters $\beta_0, \beta_1, \ldots, \beta_n$ trained on a training sample with prevalence of positive examples of τ_t instead of τ_a is then converted into the final model with parameters $\hat{\beta_0}, \hat{\beta_1}, \ldots, \hat{\beta_n}$ by correcting the intercept $\hat{\beta_0}$ as indicated in [34] :

$$\hat{\beta_0} = \beta_0 + log\frac{\tau_a}{1 - \tau_a} - log\frac{\tau_t}{1 - \tau_t}$$

The other parameters are unchanged: $\hat{\beta_i} = \beta_i$ for $i \geq 1$.

We have tried more advanced adjustments for rare events discussed in [34] (e.g., weighted logistic regression and ReLogit[6]), but the overall improvement of the quality of our predictions was only marginal.

7. Evaluation

To assess the quality of our DDI predictions, we perform two types of experiments. First, we perform a retrospective analysis that shows the ability of our system to discover valid, but yet unknown drug-drug interactions. Then, a 10-fold cross validation is performed to assess the performance of Tiresias for newly developed and existing drugs scenarios. Furthermore, we measure the prediction power of the individual local and global features and show how they affect the system performance. Finally, we evaluate the effect of adding the global embedding-based features on the performance of Tiresias and the baseline.

Hardware Setup: We deployed Tiresias on a local cluster of 8 machines. Each machine is equipped with 512GB of RAM and 4 Intel Xeon E5-4650 CPUs of 2.4GHz; 10 cores each. The machines run a 64-bit Redhat Linux and are interconnected by a 1Gbps Ethernet switch.

Implementation Details: Tiresias is written entirely in Scala. It uses Apache Spark (v 1.6) scalable machine learning library (*MLlib*) for building the logistic regression model. MLlib provides APIs that facilitates combining multiple algorithms into a single pipeline. Figure 3 shows the Spark MLLib Pipeline that Tiresias uses at training and testing. In the training phase, Tiresias, receives as input our integrated knowledge graph which includes the set of DDIs as a ground truth. This input goes to the *CrossValidator* module which is responsible for doing the data splits, calling different similarity pipelines to generate the features, building the classification model and selecting the best model. The crossValidator module selects the best of the eight models built in the model generation phase. It selects the model that produces the best F-score on the validation data. In the testing phase, Tiresias get as input pair of drugs under investigation, build their feature vectors and consults the model to check whether they interact or not. The output is the drug pair, the reference similarity features along with the probability of their interaction per feature. For the runtime of the whole process, Tiresias finishes the whole 10-fold cross validation iterations is less than 2 hours.

Datasets Statistics: Our integrated knowledge graph consists of ~160K triples representing information about 2,600 approved drugs. Tiresias uses this graph for local similarity features calculation as well as for constructing the global embedding-based similarity features using TransH [15] and HolE [16] graph embedding techniques. For word2vec-based embedding features, we used the textual versions of DrugBank and DailyMed. For DrugBank, we used July 2016 release which is approximately 500MB of drug-associated textual information. As for DailyMed, we used its full release which

[6] http://gking.harvard.edu/relogit

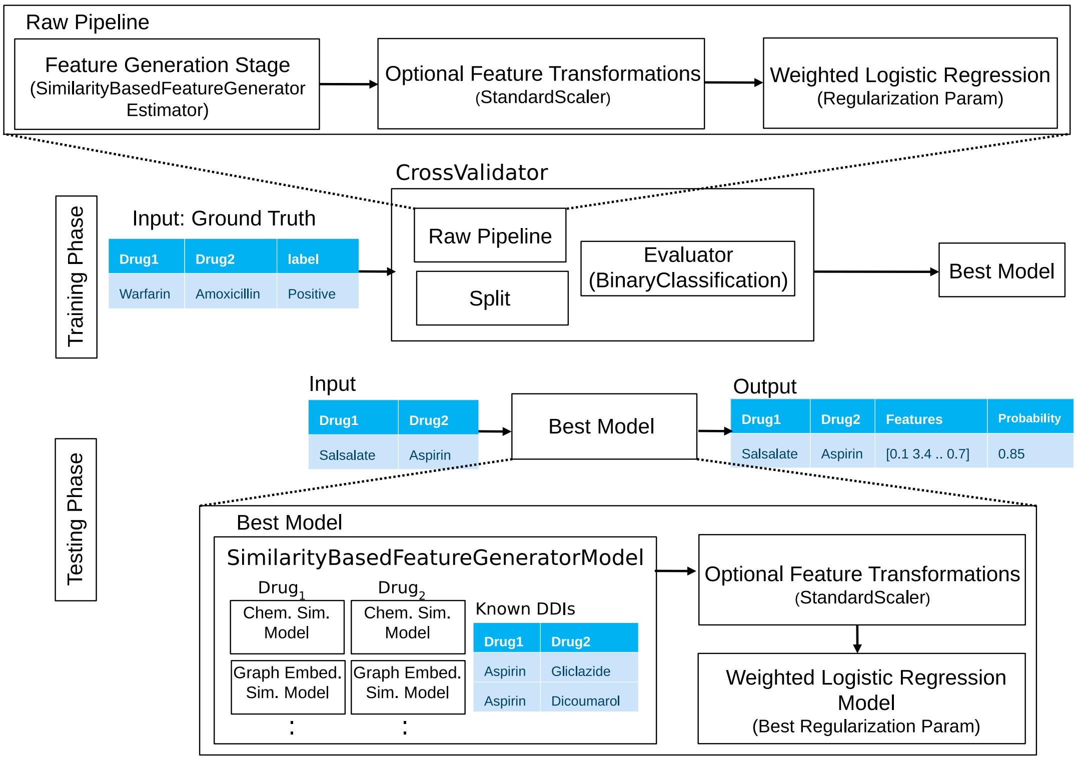


Figure 3: Tiresias Spark MLLib Pipeline

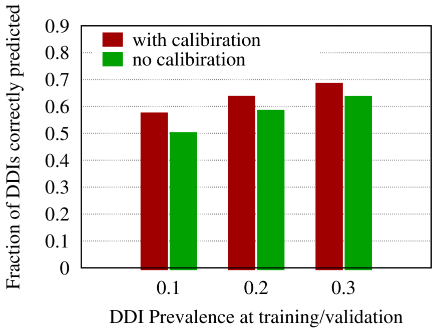


Figure 4: Retrospective Evaluation: predictions using only known DDIs as of 2011. Tiresias correctly predicts up to 68% of the DDIs found after 2011.

corresponds to around 17K parsed text files whose size is almost 2GB.

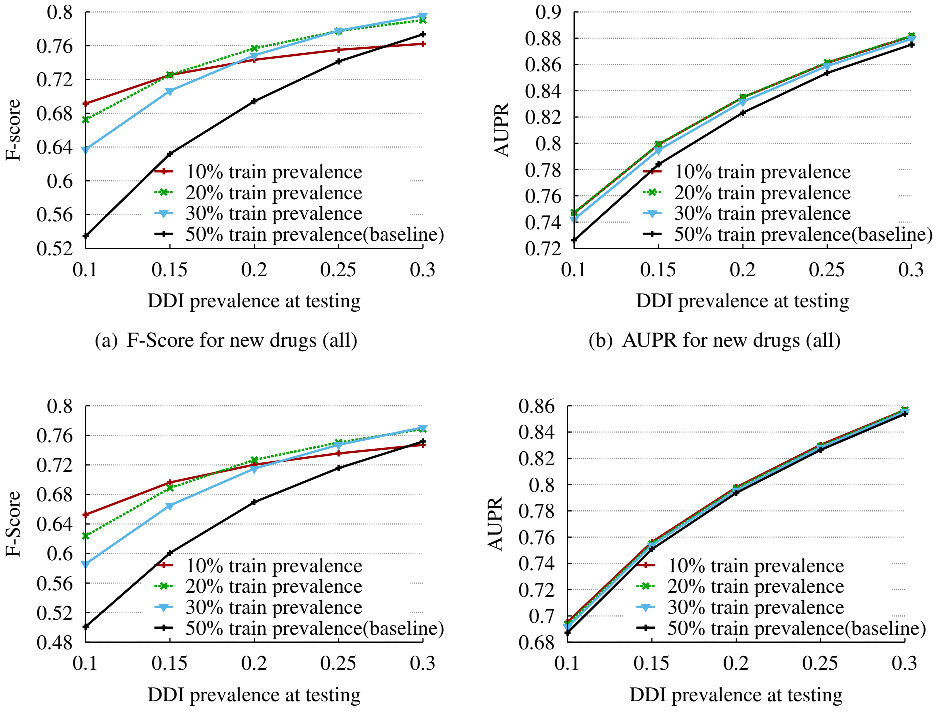
7.1. Retrospective Analysis

We perform a retrospective evaluation using as the set of known DDIs (*KDDI*) only pairs of interacting drugs present in an earlier version of DrugBank (January 2011). For different DDI prevalence at training/validation, Figure 4 shows the fraction of the total of 713 DDIs added to DrugBank between January 2011 and December 2014 that our approach can discover based only on DDIs known in January 2011 for different DDI prevalence at training/validation. Figure 4 shows that we can correctly predict up to 68% of the DDI discovered after January 2011, which demonstrates the ability of our system to discover valid, but yet unknown drug-drug interactions.

7.2. DDI Prediction Performance

In this section, we evaluate the DDI prediction performance of Tiresias for newly developed and existing drugs scenarios. We begin by first describing how the data is partitioned into training/testing followed by Tiresias prediction analysis.

Competitors: In our experiments, we compare against a baseline system which is a representative of existing similarity-based DDI prediction methods. This baseline is a version of our system that uses as input the same integrated knowledge graph and utilize the same set of local features discussed in Section 5. Notice that existing DDI prediction methods use similar features to the set of local features that we have in Tiresias (see Section 8 for details). For example, INDI [2] uses chemical-based, side-effect based and ATC-based similarities. Similarly, [7] uses chemical fingerprints while [6] uses side effects and chemical structures. Notice that the baseline assumes 50% DDI prevalence at training and it does not include the calibration features, global embedding features and the techniques for handling unbalanced data distribution. Furthermore, as shown previously in Section 1, these systems have one or more of the following problems; inability to make predictions for newly developed drugs, assuming balanced data distribution, usage of limited data sources and usage of inappropriate evaluation metrics. Therefore and as shown in the next sections, Tiresias significantly outperforms the baseline for DDIs prediction in both existing and newly developed drug scenarios.



(a) F-Score for new drugs (all)

(b) AUPR for new drugs (all)

(c) F-Score for new drugs (nocal)

(d) AUPR for new drugs (nocal)

Figure 5: Evaluating Tiresias performance for new developed drugs scenario. Using calibration features with unbalanced training/validation data, Tiresias significantly outperforms the baseline.

7.2.1. *Data Partitioning*

In the 10 fold cross evaluation of our approach, to simulate the introduction of a newly developed drug for which no interacting drugs are known, 10% of the drugs appearing as the first element of a pair of drugs in the set *KDDI* of all known drug pairs are hidden, rather than hiding 10% of the drug-drug relations as done in [2, 8, 7]. Since the drug-drug interaction relation is symmetric, we consider, without loss of generality, only drug candidate pairs (d_1, d_2) where the canonical name of d_1 is less than or equal to the canonical name of d_2 according to the lexicographic order (i.e., $d_1 \leq d_2$). In particular, pairs of drugs (d_1, d_2) in *KDDI* are such that $d_1 \leq d_2$. $Drugs_{test}$ denotes the set of hidden drugs that act as the newly developed drugs for which no DDIs are known at training or validation. This results in two subsets of *KDDI*, $KDDI_{test} = \{(d_1, d_2) | d_1 \in Drugs_{test} \wedge d_2 \notin Drugs_{test} \wedge (d_1, d_2) \in KDDI\}$ and $KDD_x = \{(d_1, d_2) | d_1 \notin Drugs_{test} \wedge d_2 \notin Drugs_{test} \wedge (d_1, d_2) \in KDDI\}$. $KDDI_{test}$ is the set of known interacting pairs to use at testing (positive examples). Likewise $KDDI_x$ is further split into validation and training set by hiding 10 % of the drugs appearing as the first element of a pair of drugs in $KDDI_x$ ($Drugs_{val}$ denotes this set of hidden drugs that act as the drugs for which no DDIs are known at training). This results in two subsets of $KDDI_x$: $KDDI_{val} = \{(d_1, d_2) | d_1 \in Drugs_{val} \wedge d_2 \notin Drugs_{val} \wedge (d_1, d_2) \in KDDI_x\}$ and $KDD_{train} = \{(d_1, d_2) | d_1 \notin Drugs_{val} \wedge d_2 \notin Drugs_{val} \wedge (d_1, d_2) \in KDDI_x\}$. $KDDI_{val}$ corresponds to the set of known interacting drugs used in the model validation phase, and $KDDI_{train}$ is the set of known interacting drugs used at training.

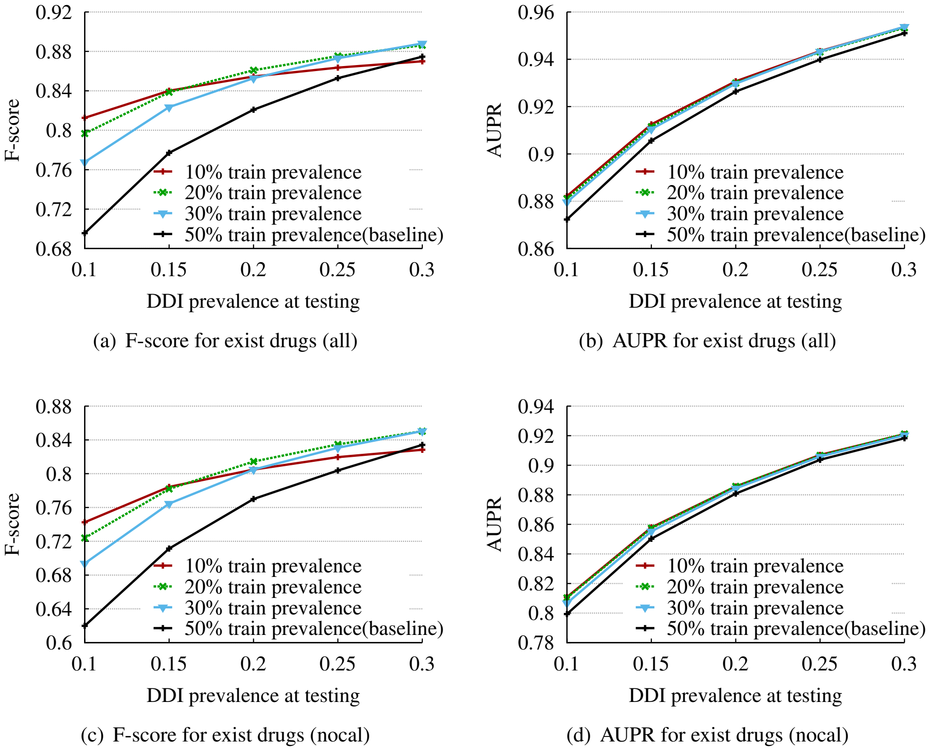
The training data set consists of (*i*) known interacting drugs in $KDDI_{train}$ as positive examples, and (*ii*) randomly generated pairs of drugs (d_1, d_2) not already known to interact (i.e, not in *KDDI*) such that the drugs d_1 and d_2 appear in $KDDI_{train}$ (as negative examples).

The validation data set consists of (*i*) the known interacting drug pairs in $KDDI_{val}$ as positive examples, and (*ii*) negative examples that are randomly generated pairs of drugs (d_1, d_2) not already known to interact (i.e., not in *KDDI*) such that d_1 is the first drug in at least one pair in $KDDI_{val}$ (i.e., a drug only seen at validation but not at training) and d_2 appears (as first or second element) in at least on pair in $KDDI_{train}$ (i.e., d_2 is known at training).

The testing data set consists of (*i*) the known interacting drug pairs in $KDDI_{test}$ as positive examples, and (*ii*) negative examples that are randomly generated pairs of drugs (d_1, d_2) not already known to interact (i.e., not in *KDDI*) such that d_1 is the first drug in at least one pair in $KDDI_{test}$ (i.e., a drug only seen at testing but not at training or validation) and d_2 appears (as first or second element) in at least on pair in $KDDI_{train} \cup KDDI_{val}$ (i.e., d_2 is known at training or at validation).

7.2.2. *DDI Prediction for Newly Developed Drugs*

Contrary to prior work, in our evaluation, the ratio of positive examples to randomly generated negative examples is not 1 to 1. Instead, the assumed prevalence of DDIs at training and validation is the same and is in the set $\{10\%, 20\%, 30\%, 50\%\}$. DDI



(a) F-score for exist drugs (all)

(b) AUPR for exist drugs (all)

(c) F-score for exist drugs (nocal)

(d) AUPR for exist drugs (nocal)

Figure 6: Evaluating Tiresias performance for existing drugs scenario. For a fixed DDI prevalence at training/validation, using calibration features is always better.

Prevalence is a quantitative measure of the percentage of DDIs existing in the dataset. For example, 20% DDI train prevalence corresponds to a training dataset with a 20% of the set of all pairs of drugs are interacting. Similarly, 10% DDI prevalence at testing means that the testing dataset has 10% of its DDIs as positive examples. For a given DDI prevalence at training and validation, we evaluate the quality of our predictions on testing data sets with varying prevalence of DDIs (ranging from 10% to 30%). 50% DDI prevalence at training and validation is used here to assess the quality of prior work (which rely on a balanced distribution of positive and negative examples at training) when the testing data is unbalanced.

For a given assumed DDI prevalence at training/validation and a DDI prevalence at testing, to get robust results and show the effectiveness of our calibration-based features, we perform not one, but five 10-fold cross validations with all the features described in section 5 (see Figures 5(a) and 5(b)) and five 10 fold-cross validations without calibration features (see Figures 5(c) and 5(d)). Results reported on Figures 5 represent average over the five 10 fold-cross validations.

The key results from our evaluation are as follows:

- Regardless of the DDI prevalence used at training and validation (provided that it is between 10% to 30% -i.e., the lower and upper bound of the true prevalence of DDIs over the set of all drug pairs), our approach using calibration features (solid lines in Figures 5(a) and 5(b)) and unbalanced training/validation data (non-black lines) significantly outperforms the baseline representing prior similarity-based approaches (e.g., [2]) that rely on balanced training data without calibration-based features (the dotted black line with crosses as markers). For an assumed DDI prevalence at training ranging from 10% to 50%, the average F-score (resp. AUPR) over testing data with prevalence between from 10% to 30% varies from 0.73 to 0.74 (resp. 0.821 to 0.825) when all features are used. However, when calibration features are not used and the training is done on balanced data, the average F-score (resp. AUPR) over testing data with prevalence between from 10% to 30% is 0.65 (resp. 0.78)[7]. The difference with the baseline is higher the skewer the testing data distribution is.

- For a fixed DDI prevalence at training/validation, using calibration features is always better in terms of F-Score or AUPR (see Figures 5(a) and 5(b) compared to 5(c) and 5(d) for the F-score and AUPR values of Tiresias with and without calibration features, respectively.)

- As pointed out in prior work, the area under ROC curves (AUROC) is not affected by the prevalence of DDI at training/validation or testing. It remains constant at about 0.92 with calibration features and 0.90 without calibration features.

[7] Precision (resp. recall) varies from 0.84 to 0.70 (resp. 0.66 to 0.78) with calibration features and unbalanced training set. Precision (resp. recall) is at 0.54 (resp. 0.84) on balanced training without calibration.

Table 1: Effect of adding embedding-based features to Tiresias at 10% DDI Prevalence at Training and Testing

		Precision	**Recall**	**F-score**	**ROC**	**AUPR**
Individual Features	Tiresias- Local Features (TLF)	0.815	0.812	0.813	0.974	0.887
	Global Features - HolE only	0.785	0.766	0.775	0.961	0.841
	Global Features - DailyMed-word2vec only	0.585	0.641	0.611	0.885	0.611
	Global Features - DrugBank-word2vec only	0.381	0.593	0.463	0.852	0.412
Combination of 2 Features	TLF + DrugBank-word2vec	0.817	0.815	0.816	0.975	0.890
	TLF + DailyMed-word2vec	0.827	0.817	0.822	0.977	0.895
	TLF + TransH	0.832	0.821	0.826	0.976	0.896
	TLF + HolE	**0.860**	**0.835**	**0.847**	**0.981**	**0.918**
Combination of 3 Features	TLF + TransH+DailyMed-word2vec	0.841	0.820	0.830	0.977	0.901
	TLF + HolE+DailyMed-word2vec	**0.871**	0.831	0.850	0.980	0.917
	TLF + HolE+DrugBank-word2vec	0.867	0.832	0.849	0.980	0.917
	TLF + HolE+DailyMed+TransH	0.866	**0.838**	**0.851**	**0.981**	**0.919**

7.2.3. DDI Prediction for Existing Drugs

We also perform 10 fold-cross validation evaluations hiding drug-drug associations instead of drugs. Figures 6(a), 6(b), 6(c) and 6(d) shows the F-score and AUPR values of Tiresias with and without calibration features, respectively. The results in Figure 6 show that, even when predictions are made only on drugs with some known interacting drugs, the combination of unbalanced training/validation data and calibration features remains superior to the baseline.

7.3. Individual Features Prediction Power

In this experiment, we test the effect of using each local feature individually on the overall performance. Our experiments show that no similarity measure by itself has a good predictive power. We found that ATC-based similarity is the best with 0.58 F-score and 0.56 AUPR. Removing any given local similarity measure has limited impact on the quality of the predictions. The greatest decrease was by 1% in the F-score and AUPR values after ATC-based similarity removal.

We also tested the prediction power of each global embedding-based similarity feature individually (see upper part of Table 1) and compare it against Tiresias using only the Local Features (TLF) discussed in Section 5. As the table shows, HolE-based graph embedding feature is the most powerful individual feature; it could achieve by its own an F-score value of 77.5%. However, HolE-based feature is still inferior compared to Tiresias using all local features which has an F-Score of 81.3%. Moreover, word2vec embeddings; either based on DailyMed or DrugBank, did not perform very well by itself compared to HolE or the local features. Consequently, we believe that the good prediction performance of Tiresias is a result of combining all the features together and not by any individual feature (see Section 7.4).

7.4. Combining Local and Global Embedding Features

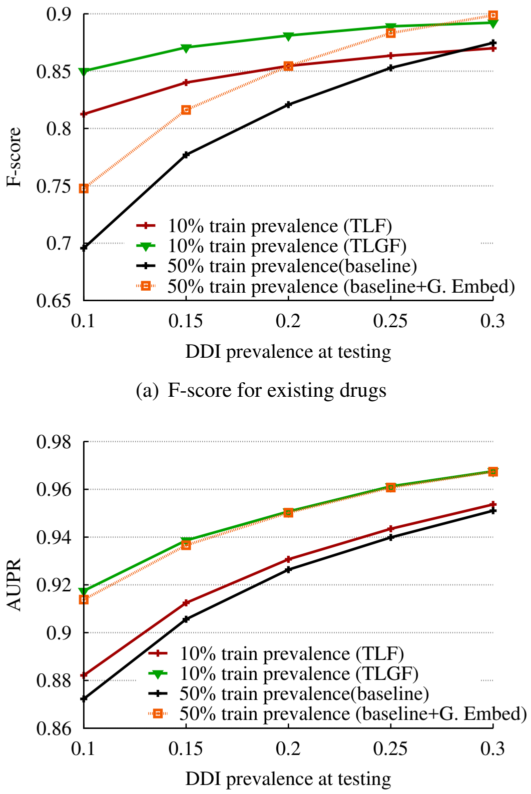
In this experiment, we measure the effect of adding the global embedding-based features to Tiresias. These features include both word and graph embedding features as discussed in Section 5. Specifically, we compare against TLF which is Tiresias using all our supervised local features. Table 1 shows the effect of adding each embedding-based feature individually followed by combining multiple features together. The word2vec features results a modest improvement over TLF F-score performance by 0.3% and 1% for DrugBank and DailyMed, respectively. On the other hand, using graph embedding based features improved the performance of TLF significantly. TransH improved the F-score value of the baseline by 1.3% while HolE increased that value by almost 3%. We also tested the effect of combining these features together. The lower part of the table shows Tiresias performance when we combine word and graph embedding features together which generally shows better performance than using these features individually. The best performance is obtained when we combined DailyMed word2vec with HolE and TransH graph embedding which gains almost a 4% F-score improvement over the Tiresias using only local features.

We also show how the DDI prevalence ratio affects the performance of Tiresias when using the embedding features. In this experiment, we use all supervised features in addition to DailyMed word2vec and HolE graph embedding. This version is coined Tiresias using both Local and Gocal Features (TLGF). Figure 7 shows how Tiresias performance varies as we change the DDI prevalence at training and testing. We use the lower- and upper-bound values of DDI prevalence at training used in Figures 5 and 6 which are 10% and 50%, respectively. As Figure 7 demonstrates, introducing the embedding features improves the performance of both TLF as well as the baseline significantly. For example, at 10% DDI at testing prevalence, the F-score of Tiresias using both local and global features (TLGF) increased from 0.81 to 0.85 (resp. AUPR increased from 0.88 to 0.91). Similarly, the F-score value of the baseline improved from 0.69 to 0.74 while AUPR improved from 0.87 to 0.91.

7.5. Discussion

In this study, we proposed Tiresias; a large-scale system for predicting DDIs. Our results show that Tiresias is effective in predicting new interactions among existing as well as newly developed drugs. We summarize below our main findings:

- For predicting DDIs among newly developed drugs and using only known DDIs before 2011 for training, Tiresias



(a) F-score for existing drugs

(b) AUPR for existing drugs

Figure 7: Effect of DDI prevalence ratio on Tiresias when using local and global embedding-based similarity features. At the lower and upper values for training prevalence, embedding features improves the performance of both the baseline as well as TLF significantly.

was able to correctly predict 68% of the DDIs found after 2011.

- We have also shown how the assumption of a balanced distribution of interacting drug pairs severely affects the performance. This shows the importance and the effectiveness of the proposed calibration features and the techniques we proposed for handling unbalanced datasets.

- Our experiments show that Tiresias outperforms existing systems by up to ∼11% for the existing drugs scenario and by up to ∼15% for the newly develop drug scenario.

- We also show that the high prediction power of Tiresias comes from the combination of the various proposed local and global similarity features and not by any individual feature.

- Finally, we show that our newly introduced global embedding-based similarity features could enhance the performance further by an increase of 4% and 5% over the F-score value of Tiresias using local features only (TLF) and the baseline, respectively.

In the current version of Tiresias, we only predicted whether two drugs interact or not without providing further information about the type of interaction. In future work, we plan to extend our model to give detailed predictions of the nature, severity, and cause of DDIs. We also plan to investigate the possibility of identifying subsets of features that characterize DDIs of certain groups of drugs.

8. Related Work

In this section, we briefly review existing computational approaches for predicting DDIs. We discuss mainly feature vector-based and similarity-based DDI prediction methods. Notice that all the methods discussed in this section have one or more of the following shortcomings (see Section 1 for details); (*i*) Inability to make predictions for newly-developed drugs. (*ii*) Ignoring the skewed distribution of interacting drug pairs. (*iii*) Discarding many relevant data sources and (*iv*) usage of inappropriate evaluation metrics.

8.1. Direct feature vector-based approaches

The inputs of general machine learning methods are instances, which can be represented by feature vectors. In our setting, instances are pairs of drugs, and their feature vectors can be generated by directly combining features of two drugs (e.g., chemical descriptors of two drugs). With these inputs, any standard machine learning method (e.g., logistic regression, support vector machines) can be used to build models for predicting drug-drug interactions. Luo et al [3], for example, proposes a feature vector-based DDI prediction server that makes real-time DDI predictions based only on molecular structure. Given the molecular structure of a drug d, the server docks it across 611 human proteins to calculate a docking score of the molecule to each human protein target. This produces a 611-dimensional docking vector $v(d)$. The feature vector associated with a pair of drugs (d_1, d_2) is then computed as the concatenation of the two vectors $v(d_1)+v(d_2)$ and $|v(d_1)-v(d_2)|$ to produce a 1222-dimensional vector (here, for a vector x, $|x|$ denotes the vector obtained by taking the absolute value of each component of x). Finally, a logistic regression model is built based on these features for DDI predictions. The model can suggest potential DDIs between a user's molecule and a library of 2515 drug molecules.

8.2. Similarity-based DDI prediction approaches

Similarity-based approaches [2, 6, 7] try to predict whether a candidate pair of drugs interacts by comparing it against known interacting pairs of drugs. Finding known interacting drugs that are very similar to the candidate pair provides supporting evidence for the existence of a drug-drug interaction between the two candidate drugs. INDI (INferring Drug Interactions) [2] has three phases; construction of drug-drug similarity measures, building classification features and applying a classifier to predict new DDIs. INDI used seven drug-drug similarity measures including chemical similarity, similarities based on registered and predicted side effects, the Anatomical, Therapeutic and Chemical (ATC) classification system and three similarity measures constructed between drug targets. These features are then combined into 49 features to calculate the maximum similarity between the query drug pair and all the known DDIs existing in the database. Vilar et al. [7] proposed a similarity-based modeling protocol that uses several similarity measures to predict novel DDIs. These measures include structure similarity, interaction profile fingerprints, 3D pharmacophoric similarity, drug-target similarity and adverse drug effects similarity. This proposed protocol is a multi-type predictor that can isolate the pharmacological or clinical effect associated with the predicted interactions. Zhang et al. [6] proposed an integrative label propagation framework to predict DDIs. It integrates multiple similarity measures together including side effects extracted from prescription drugs, side effects extracted from FDA Adverse Event Reporting System, and chemical structures from PubChem. In addition to predicting DDIs, their proposed method is also able to rank drug information sources based on their contributions to the prediction.

9. Conclusion

In this paper, we proposed Tiresias; a large-scale computational framework that predicts DDIs through similarity-based link prediction. Tiresias addresses the limitations of existing approaches by: (*i*) utilizing information from various data sources, (*ii*) using larger set of local and global similarity features, (*iii*) handling data skewness and similarity measures incompleteness and (*v*) being able to make DDI predictions for existing drugs as well as newly developed drugs. We extensively evaluated Tiresias using real datasets to assess its performance. Experimental results clearly show the effectiveness of Tiresias in both predicting new interactions among newly developed and existing drugs. It also shows that the combination of locally and globally generated drugs similarity features improves the performance of Tiresias significantly. The predictions provided by Tiresias will help clinicians to avoid hazardous DDIs in their prescriptions and will aid pharmaceutical companies to design large-scale clinical trials by assessing potentially hazardous drug combinations.

10. References

[1] D. Flockhart, P. Honig, S. Yasuda, C. Rosebraugh, Preventable adverse drug reactions: A focus on drug interactions, Centers for Education & Research on Therapeutics 452.

[2] A. Gottlieb, G. Y. Stein, Y. Oron, E. Ruppin, R. Sharan, Indi: a computational framework for inferring drug interactions and their associated recommendations, Molecular systems biology 8 (1) (2012) 592.

[3] H. Luo, P. Zhang, H. Huang, J. Huang, E. Kao, L. Shi, L. He, L. Yang, Ddi-cpi, a server that predicts drug-drug interactions through implementing the chemical-protein interactome, Nucleic Acids Research 42 (2014) W46–W52.

[4] P. Zhang, P. Agarwal, Z. Obradovic, Computational drug repositioning by ranking and integrating multiple data sources, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2013, pp. 579–594.

[5] P. Zhang, F. Wang, J. Hu, R. Sorrentino, Towards personalized medicine: Leveraging patient similarity and drug similarity analytics, AMIA Summits on Translational Science Proceedings 2014 (2014) 132.

[6] P. Zhang, F. Wang, J. Hu, R. Sorrentino, Label propagation prediction of drug-drug interactions based on clinical side effects, Scientific reports 5 (2015) 12339.

[7] S. Vilar, E. Uriarte, L. Santana, T. Lorberbaum, G. Hripcsak, C. Friedman, N. P. Tatonetti, Similarity-based modeling in large-scale prediction of drug-drug interactions, Nature protocols 9 (9) (2014) 2147–2163.

[8] S. Vilar, E. Uriarte, L. Santana, N. P. Tatonetti, C. Friedman, Detection of drug-drug interactions by modeling interaction profile fingerprints, PloS one 8 (3) (2013) 1–11.

[9] J. Davis, M. Goadrich, The relationship between precision-recall and roc curves, in: Proceedings of the 23rd international conference on Machine learning, ACM, 2006, pp. 233–240.

[10] A. Fokoue, M. Sadoghi, O. Hassanzadeh, P. Zhang, Predicting drug-drug interactions through large-scale similarity-based link prediction, in: International Semantic Web Conference, Springer, 2016, pp. 774–789.

[11] A. Fokoue, O. Hassanzadeh, M. Sadoghi, P. Zhang, Predicting drug-drug interactions through similarity-based link prediction over web data, in: Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume, 2016, pp. 175–178.

[12] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[14] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems, 2013, pp. 2787–2795.

[15] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: AAAI, Citeseer, 2014, pp. 1112–1119.

[16] M. Nickel, L. Rosasco, T. Poggio, Holographic embeddings of knowledge graphs, arXiv preprint arXiv:1510.04935.

[17] C. Knox, V. Law, T. Jewison, P. Liu, S. Ly, A. Frolkis, A. Pon, K. Banco, C. Mak, V. Neveu, et al., Drugbank 3.0: a comprehensive resource for 'omics' research on drugs, Nucleic acids research 39 (suppl 1) (2011) D1035–D1041.

[18] A. P. Davis, C. G. Murphy, C. A. Saraceni-Richards, M. C. Rosenstein, T. C. Wiegers, C. J. Mattingly, Comparative toxicogenomics database: a knowledgebase and discovery tool for chemical–gene–disease networks, Nucleic acids research 37 (suppl 1) (2009) D786–D792.

[19] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, et al., Uniprot: the universal protein knowledgebase, Nucleic acids research 32 (suppl 1) (2004) D115–D119.

[20] A. Chatr-aryamontri, B.-J. Breitkreutz, R. Oughtred, L. Boucher, S. Heinicke, D. Chen, C. Stark, A. Breitkreutz, N. Kolas, L. O'Donnell, et al., The BioGRID interaction database: 2015 update, Nucleic acids research 43 (D1) (2015) D470–D478.

[21] O. Bodenreider, The unified medical language system (UMLS): integrating biomedical terminology, Nucleic acids research 32 (suppl 1) (2004) D267–D270.

[22] C. E. Lipscomb, Medical subject headings (mesh), Bulletin of the Medical Library Association 88 (3) (2000) 265.

[23] S. H. Brown, P. L. Elkin, S. Rosenbloom, C. Husser, B. Bauer, M. Lincoln, J. Carter, M. Erlbaum, M. Tuttle, VA National Drug File Reference Terminology: a cross-institutional content coverage study, Medinfo 11 (Pt 1) (2004) 477–81.

[24] M. Sadoghi, K. Srinivas, O. Hassanzadeh, Y. Chang, M. Canim, A. Fokoue, Y. A. Feldman, Self-curating databases, in: Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016., 2016, pp. 467–472.

[25] A. Chandel, O. Hassanzadeh, N. Koudas, M. Sadoghi, D. Srivastava, Benchmarking declarative approximate selection predicates, in: ACM SIGMOD International Conference on Management of Data, SIGMOD '07, 2007, pp. 353–364.

[26] M. Kuhn, M. Campillos, I. Letunic, L. J. Jensen, P. Bork, A side effect resource to capture phenotypic effects of drugs, Molecular systems biology 6 (1) (2010) 343.

[27] P. Resnik, et al., Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language, J. Artif. Intell. Res.(JAIR) 11 (1999) 95–130.

[28] K. Ovaska, M. Laakso, S. Hautaniemi, Fast gene ontology based clustering for microarray experiments, BioData mining 1 (1) (2008) 11.

[29] C. Steinbeck, C. Hoppe, S. Kuhn, M. Floris, R. Guha, E. L. Willighagen, Recent developments of the chemistry development kit (cdk)-an open-source java library for chemo-and bioinformatics, Current pharmaceutical design 12 (17) (2006) 2111–2120.

[30] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, E. Willighagen, The chemistry development kit (cdk): An open-source java library for chemo-and bioinformatics, Journal of chemical information and computer sciences 43 (2) (2003) 493–500.

[31] A. Skrbo, B. Begović, S. Skrbo, [classification of drugs using the atc system (anatomic, therapeutic, chemical classification) and the latest changes]., Medicinski arhiv 58 (1 Suppl 2) (2003) 138–141.

[32] G. Sidorov, A. Gelbukh, H. Gómez-Adorno, D. Pinto, Soft similarity and soft cosine measure: Similarity of features in vector space model, Computación y Sistemas 18 (3) (2014) 491–504.

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

[34] G. King, L. Zeng, Logistic regression in rare events data, Political Analysis 9 (2) (2001) 137–163.