

# Predicting Polypharmacy Side-effects Using Knowledge Graph Embeddings

Vít Nováček<sup>1,2</sup>, Sameh K. Mohamed<sup>1,2</sup>

<sup>1</sup>Data Science Institute, NUI Galway

<sup>2</sup>Insight Centre for Data Analytics, NUI Galway

## Abstract

Polypharmacy is the use of drug combinations and is commonly used for treating complex and terminal diseases. Despite its effectiveness in many cases, it poses high risks of adverse side effects. Polypharmacy side-effects occur due to unwanted interactions of combined drugs, and they can cause severe complications to patients which results in increasing the risks of morbidity and leading to new mortalities. The use of drug polypharmacy is currently in its early stages; thus, the knowledge of their probable side-effects is limited. This encouraged multiple works to investigate machine learning techniques to efficiently and reliably predict adverse effects of drug combinations. In this context, the Decagon model is known to provide state-of-the-art results. It models polypharmacy side-effect data as a knowledge graph and formulates finding possible adverse effects as a link prediction task over the knowledge graph. The link prediction is solved using an embedding model based on graph convolutions. Despite its effectiveness, the Decagon approach still suffers from a high rate of false positives. In this work, we propose a new knowledge graph embedding technique that uses multi-part embedding vectors to predict polypharmacy side-effects. Like in the Decagon model, we model polypharmacy side effects as a knowledge graph. However, we perform the link prediction task using an approach based on tensor decomposition. Our experimental evaluation shows that our approach outperforms the Decagon model with 12% and 16% margins in terms of the area under the ROC and precision recall curves, respectively.

## Introduction

Polypharmacy side-effects are a specific case of adverse drug reactions that can cause significant clinical problems and represent a major challenge for public health and pharmaceutical industries <sup>(1)</sup>. Pharmacology profiling leads to identification of both intended (target) and unintended (off-target) drug-induced effects, i.e. biological system perturbations. While most of these effects are discovered during pre-clinical and clinical trials before a drug release on the market, some potentially serious adverse effects only become known when the drug already is in use. Indeed, a recent review of epidemiological studies in Europe, states that 3.5% of hospitalisation admissions are due to adverse drug reactions and 10% of patients experience an adverse drug reaction during their hospitalisation <sup>(2)</sup>. An adverse drug reaction is a major cause of morbidity (resulting in reduction of patients' quality of life) and mortality <sup>(3,4)</sup>. Recent estimates set the number of yearly drug-induced fatalities to 100,000 in the USA and almost 200,000 in Europe, making it the fourth cause of death before pulmonary disease or diabetes <sup>(5,2)</sup>.

When more drugs are used jointly (i.e. polypharmacy), the risk of adverse effects rises rather rapidly <sup>(6,7)</sup>. Therefore, reliable automated prediction of such risks are highly desirable to mitigate their impact on patients. Among the recent approaches is the Decagon model <sup>(8)</sup> which has achieved state-of-the-art performance by modelling the data related to polypharmacy side-effects as a knowledge graph and formulating finding possible adverse effects as a link prediction task over the knowledge graph.

The link prediction is solved using an embedding model based on graph convolutions in the Decagon model <sup>(8)</sup>. Despite its effectiveness, this approach still suffers from a high rate of false positives. Here, we propose a related knowledge graph embedding technique that, instead, uses multi-part embedding vectors to predict polypharmacy side-effects. Contrary to the Decagon model <sup>(8)</sup>, the link prediction problem is solved using an approach based on tensor decomposition. This is known to lead to superior performance in the general link prediction problem <sup>(9,10)</sup>.

Our approach models drug pairs and their side-effects as a knowledge graph, where interacting drugs modelled as nodes and their corresponding side-effects are modelled as edges. We then use a tensor factorisation based knowledge graph embedding model to learn vector representations of both the drugs and side-effects. Finally, the learnt embeddings are used to predict unknown drug–drug interactions polypharmacy side-effects. Our experimental evaluation shows that this approach outperforms the Decagon model with 12% and 16% margins in terms of the area under the ROC and precision-recall curves, respectively.

The rest of the paper is organised as follows: Background and related work section provides reviews on the basic notions related works to the addressed problem. Data section comments on the data used. The proposed method is detailed in the methods section. Experiments and Results and discussion sections describe the experiments we performed and discuss the outcome results. The paper is concluded in in the conclusions section where we also discuss the future directions of our research.

## Background and related work

In this section, we present a set of preliminary topics and concepts that are used across this study. We present the knowledge graph as a technique to model linked data, and we also present knowledge graph embedding models in the context of learning link prediction on knowledge graphs.

**Knowledge graph embedding.** Knowledge graphs are structured knowledge bases which are used to model linked data. They model knowledge entity as graph nodes and relations between them are modelled with graph edges. In recent years, knowledge graphs have become a popular means for data representation in the semantic web community to create the "web of data", which is a network of interconnected entities that can be easily interpreted by both humans and machines <sup>(11)</sup>. They have also been used as a convenient means for modelling information in many different domains, including general human knowledge <sup>(12)</sup>, biomedical information <sup>(13)</sup> and language lexical information <sup>(14)</sup>. Knowledge graphs are now used in different applications such as enhancing semantics of search engine results <sup>(15, 16)</sup>, biomedical discoveries <sup>(17)</sup>, or powering question answering and decision support systems <sup>(18)</sup>.

Knowledge graphs model facts that are formatted as (subject, predicate, object) triples such as (drug<sub>*i*</sub>, interacts-with, protein<sub>*j*</sub>), where the subject and object are entities and the predicate is a relation between them.

Knowledge graph embedding models are representation learning models that embed components of the knowledge graph including entities and relations into a low-dimensional continuous vector space. In the context of link prediction, these embeddings are then used to rank triples according to their factuality. In recent years, knowledge graph embedding models have witnessed multiple developments that allowed them to excel in the task of link prediction over knowledge graphs <sup>(19)</sup>, where they are known to provide state-of-the-art predictive accuracy <sup>(9)</sup>.

Knowledge graph embedding models operate in a multi-phase procedure. Initially, they consume input triples and initial embeddings of entities and relations. They then use an iterative optimisation routine to update these embeddings until they reach a local optima state (according to training iterations and parameters). In practice, training data is divided into batches and the optimisation routine is applied on each batch independently. First, the knowledge graph embedding model generates negative training samples, where this procedure typically uses uniform random corruptions of the subjects and objects of the training triples. The model then looks up the corresponding embeddings of the components (subject, predicate and object) of both the true and corrupted triples. The embeddings are then used to compute a score for each triple using a model-dependant scoring function. The model then computes a training loss that models the difference between the scores of true and corrupted triples. Finally, the loss function is used to compute gradient updates to the knowledge component embeddings in order to maximise the scores of true triples and minimise the scores of negative samples.

Different knowledge graph embeddings models vary in terms of the scoring functions and loss objectives *i.e.* the loss function they utilise. For example, the Translating Embedding (TransE) model <sup>(20)</sup> model the interactions of triple embeddings in its scoring function as a translation from the subject vector to the object vector using the relation vector in the embedding space. On the other hand, other models such as the RESCAL <sup>(21)</sup>, DistMult <sup>(22)</sup> and ComplEx <sup>(23)</sup> models learn scores using the product of the embeddings of the investigated triples components. Other models such as the Graph Convolution Network model (GCN) <sup>(24)</sup>—which is the base work of the Decagon model <sup>(8)</sup>—models embedding interactions using a graph convolutional filter on the embeddings of the triples components and their neighbouring entities.

In the rest of this paper, we use  $\mathcal{E}$  and  $\mathcal{R}$  to denote the set of all entities and relations in a knowledge graph respectively, where  $N_e$  and  $N_r$  represent the number of instances in  $\mathcal{E}$  and  $\mathcal{R}$  respectively. We also use  $\Theta_E$  and  $\Theta_R$  which denote the embeddings of entities and relations respectively, where  $\Theta_E(i)$  is the embedding of entity  $i$ ,  $\Theta_R(j)$  is the embedding of relation  $j$ , and  $f_m(s, r, o, \Theta)$  denotes the score of the fact that a subject entity  $s$  is connected to an object entity  $o$  with a

relation  $r$  based on the embedding values  $\Theta$  of the model  $m$ .

**Link prediction on knowledge graphs.** In recent years, various predictive frameworks were developed to predict new links in knowledge graphs, where these frameworks serve in various applications such as semantic search engines (15, 16), biomedical discoveries (17), and question answering systems (18). Link prediction models can be categorised into two categories: graph-feature based models and latent-feature based models.

Graph-feature based models utilise graph features like paths and graph patterns to predict possible connecting links between graph entities. For example, the path ranking algorithm (PRA) (25) uses connecting paths between entities generated by random walks to infer possible links between them. Where as other models like the subgraph feature extraction model (SFE) (26) and the distinct subgraph path (DSP) (27) employ a combination of connecting path and subgraph paths of two entities to predict their possible associations.

On the other hand, latent-feature based models *i.e.* knowledge graph embedding models, use a generative approach to learn low-rank embeddings for knowledge entities and relations in order to score their possible associations. These approaches use multiple techniques like tensor factorisation (as in the DistMult model (20)) and latent distance similarity (as in the TransE model (22)) to model possible interactions between graph embeddings and provide scores for possible graph links. For further information on both approaches, Nickel et. al. (28) provides an extended review for both graph-feature based and latent-feature based models in the task of link prediction in knowledge graphs.

**Modelling polypharmacy data.** Computation methods have been widely used to model drugs' associations to target proteins (29, 30, 31) in order to identify potential new drug therapeutic uses and new possible side-effects. In addition, other types of models were developed to directly learn the associated adverse side-effects of drugs (32, 33). Despite the success of such models, their objective of these methods was, however, limited to the side-effects that occur due to the use of individual drugs.

The use of drug combinations *i.e.* polypharmacy is known to be an important solution for complex and terminal diseases (34, 35). However, the risk of adverse effects in polypharmacy rises rather rapidly (6, 7) compared to the use of individual drugs due to unwanted drug-drug interactions. This encouraged research in developing computational methods to predict the possibility of interactions between combinations of drugs (36, 37, 38). Although these works were able to provide predictions for unknown drug-drug interactions, they were limited to predicting the existence of such interactions and not their associated side-effects. Recently, Zitnik et. al. (8) introduced the first method for predicting polypharmacy side-effects, the Decagon model, where they modelled the polypharmacy side-effects data as a knowledge graph. They then transformed the problem of predicting polypharmacy side-effects to a link prediction task on a knowledge graph, where they applied graph convolution network embedding models to predict new drug combination side-effects. In a recent work, Malone et. al. (39) suggested using other knowledge graph embedding models such as the DistMult (26) and  $KB_{LRN}$  (40) models, where they have shown that these models are capable of outperforming the Decagon model in predicting polypharmacy side-effects.

## Data

In this study, we use the dataset compiled by Zitnik et al. (8) which includes information about multiple polypharmacy drug side-effects<sup>1</sup>. The dataset also contains facts about single drug side-effects, protein-protein interactions and protein-drug targets. In the following, we discuss and summarise the content of this dataset.

**Side-effects data.** The drug side-effects represented in the dataset are collected from the SIDER (Side Effect Resource) database (41) and the OFFSIDES and TWOSIDES databases (7). These side-effects are then categorised into two groups: mono-drug and polypharmacy drug-drug interaction side-effects. The mono drug side-effects are the side effects observed from the use of a single drug from the OFFSIDES and SIDER database, while the polypharmacy side-effects are the ones that are associated with the interaction of drug pairs from the TWOSIDES and SIDER databases.

We have also use the same data splitting procedure introduced by Zitnik et. al. (8) where we have separated the side-effects in dataset into two groups according to their coverage, The group of highly represented polypharmacy

---

<sup>1</sup><http://snap.stanford.edu/decagon/>

side-effects (more than 500 associated drug combinations) is considered in for evaluation and testing, while other low represented side-effects are considered only for the training of the models.

**Protein and drug interactions data.** The investigated dataset contains both protein-protein interactions and drug-protein interactions. The protein-protein interactions represent a collection of physical interactions experimentally assessed in humans and compiled from different public sources (42, 43, 44, 45). On the other hand, the drug-protein interactions are obtained from the STITCH database (44). These interactions denote the associations between drugs and their target proteins in the human body that allow the drugs to activate their expected therapeutic effects and their other unwanted effects (side-effects).

## Methods

In the following, we discuss the different components of our predictive method and how it works.

**Embeddings representation.** The knowledge graph embedding model that we use follow a generative approach to learn low-rank embedding vectors for knowledge entities and relations. For learning the embeddings, multiple techniques can be used, such as tensor factorisation (c.f. the DistMult model (20)) or latent distance similarity (c.f. the TransE model (22)). The goal of all these techniques is to model possible interactions between graph embeddings and to provide scores for possible graph links. In the following, we provide details on the knowledge graph embedding procedure and the design of our model, the TriVec model.

The TriVec model is a tensor factorisation based knowledge graph embedding model that extends the works of the DistMult (22) and ComplEx (23) models. It represents each entity and relation using three embedding vectors such that the embedding of entity  $i$  is  $\Theta_E(i) = \{e_i^1, e_i^2, e_i^3\}$  where all embedding vectors have the same size  $K$  (a user-defined embeddings size). Similarly, the embedding of relation  $j$  is  $\Theta_R(j) = \{w_j^1, w_j^2, w_j^3\}$ .  $e^m$  and  $w^m$  denote the  $m$  part of the embeddings of the entity or the relation, and  $m \in \{1, 2, 3\}$  represents the three embeddings parts. The embeddings in the TriVec model are initially with random values generated by the Glorot uniform random generator (46). The embedding vectors are then updated during the training procedure to provide optimised scores for the knowledge graph facts.

**Training procedure.** The TriVec is a knowledge graph embedding model that follows the multi-phase procedure discussed in (Section ) to effectively learn a vector representation for entities and relations of a knowledge graph. First, the model initialises its embeddings with random noise. It then updates them by iterative learning on the training data. In each training iteration *i.e.* epoch, the model splits the training data into mini-batches and executes its learning pipeline over each batch. The learning pipeline of the model learns the embeddings of entities and relations by minimising a negative softmax log-loss that maximises the scores of true facts and minimises the scores of unknown facts (assumed false during training). This loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{spo}^{\text{TriModel}} = & -\phi_{spo} + \log\left(\sum_{o'} \exp(\phi_{spo'})\right) \\ & -\phi_{spo} + \log\left(\sum_{s'} \exp(\phi_{s'po})\right) \\ & + \frac{\lambda}{3} \sum_{k=1}^K \sum_{m=1}^3 (|e_s^m|^3 + |w_p^m|^3 + |e_o^m|^3) \end{aligned} \quad (1)$$

where  $K$  denotes the embedding size,  $\phi_{spo}$  denotes the score of the  $(s, p, o)$  triple,  $x'$  represents an entity  $e : e \neq x, e \in \mathcal{E}$ ,  $e_i^m$  is the embedding part  $m$  of the entity embedding  $\Theta_E(i)$ ,  $w_i^m$  is the embedding part  $m$  of the relation embedding  $\Theta_R(i)$ ,  $\phi_{spo}$  denotes the score of the triple  $(s, p, o)$ ,  $m$  denotes the embedding part index,  $\lambda$  denotes a configurable regularisation weight parameter and  $|x|$  is the absolute of  $x$ . The term  $\frac{\lambda}{3} \sum_{k=1}^K \sum_{m=1}^3 (|e_s^m|^3 + |w_p^m|^3 + |e_o^m|^3)$  is the nuclear 3-norm, which is a regularisation term (9) that enhances model generalisation over datasets with large entity vocabularies.

The scores of the TriVec model are computed using an embeddings interaction function (scoring function) that is defined

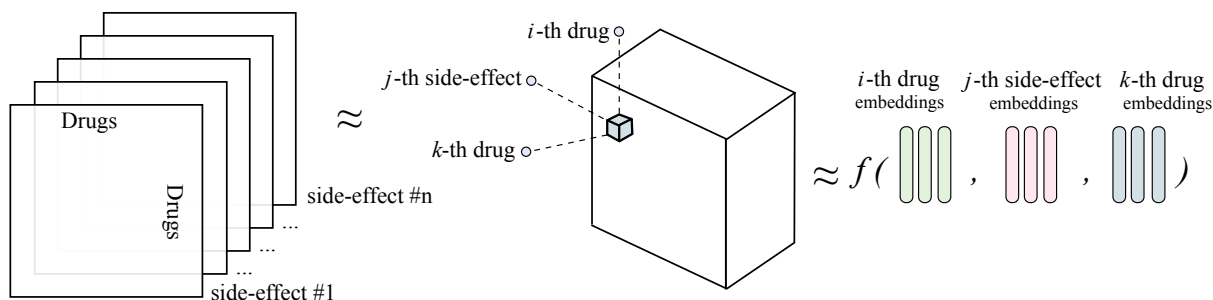


Figure 1: Flow diagram of the scoring function of TriModel applied to polypharmacy side-effects drug combinations. The subject (s), and the object (o) represent the drug combination while their corresponding polypharmacy side-effect is represented as the relation (r). All these components are represented using three embedding vectors of size  $k$ . The score of a triple  $(s, p, o)$  is defined as  $f(s, p, o) = \sum_k c_k$ , where  $c = i_1 + i_2 + i_3$ ,  $i_1 = s_1 \cdot r_1 \cdot o_3$ ,  $i_2 = s_2 \cdot r_2 \cdot o_2$ , and  $i_3 = s_3 \cdot r_3 \cdot o_1$ .

as follows:

$$f_{\text{TriModel}}(s, r, o, \Theta) = \sum^K e_s^1 w_r^1 e_o^3 + e_s^2 w_r^2 e_o^2 + e_s^3 w_r^3 e_o^1. \quad (2)$$

It uses a set of three interactions: one symmetric interaction:  $(e_s^2 w_r^2 e_o^2)$  and two asymmetric interactions:  $(e_s^1 w_r^1 e_o^3)$  and  $(e_s^3 w_r^3 e_o^1)$ . This approach models both symmetry and asymmetry in simple form similar to the DistMult<sup>(22)</sup> model where the DistMult model can be seen as a special case of the TriVec model if the first and third embeddings parts are equivalent ( $e^1 = e^3$ ).

**Modelling polypharmacy side-effects.** The TriVec model is a tensor factorisation based embedding model that solves the problem of link prediction as a 3D tensor completion, where the tensor dimensions represent entities and relations. In the task of predicting polypharmacy side-effects, the drug combinations are modelled as the subjects and objects of triples while the corresponding polypharmacy side-effects are modelled as relations. In the training process, the model processes the different types of assertions such as protein-protein interactions, drug-protein interactions, drug-drug interactions, single drug side effects and polypharmacy side-effects. This allows the model to learn efficient embeddings for the components corresponding to the different entities and relations in the knowledge graph. In the prediction phase, the TriVec model then learns the probability of polypharmacy side-effects associations to drug combinations by completing a 3D tensor of drugs and polypharmacy side-effects as shown in (Figure 1).

## Experiments

In this section, we discuss the setup of our experiments, the evaluation protocol and the detail of the frameworks and technologies used to implement our experiments.

**Benchmarking dataset.** In this study, we build a benchmarking dataset to evaluate our model following the evaluation protocol proposed by Zitnik et. al.<sup>(8)</sup>. We first divide polypharmacy side-effects assertions into groups according to the side-effect type. We then divide the assertions of each group into three groups: training, validation and testing with 80%, 10% and 10% percentages of the data respectively. This process is applied to the polypharmacy side-effect groups with 500 or more assertions to assure a minimum of 50 validation and testing instances for each side-effect. We then add other types of assertions such as drug-protein interactions, protein-protein interactions and single drug side-effects into the training data.

For each validation and testing splits, we generate negative samples by using random unobserved drug combinations as negatives. This process is executed independently for each polypharmacy side-effect, where the positive to negative ratio is 1:1. (Table 1) shows a summary of the statistics of different components in the data generated for the set of all polypharmacy side-effects.

Table 1: Summary of statistics of entities, relations and triples in the different splits of the benchmarking dataset.

Dataset	Entities	Relations	Triples	P. Side-effects
Training data	32K	967	4.7M	3.7M
Validation data	643	963	459K	459K
Testing data	643	963	459K	459K
All	32K	967	5.6M	4.6M

**Experimental setup.** We use the supporting knowledge graph to perform a grid search to learn the model’s best hyperparameters. In all of our experiments we initialise our model’s embeddings using the Glorot uniform random generator<sup>(46)</sup> and we optimise the training loss using the Adagrad optimiser, where the learning rate ( $lr$ )  $\in \{0.1, 0.01, 0.001\}$ , embeddings size ( $K$ )  $\in \{50, 100, 150, 200\}$  and batch size ( $b$ )  $\in \{512, 1024, 4000, 6000\}$ . The rest of the grid search hyperparameters are defined as follows: the regularisation weight ( $\lambda$ )  $\in \{0.1, 0.3, 0.35, 0.01, 0.03, 0.035\}$  and dropout ( $d$ )  $\in \{0.0, 0.1, 0.2, 0.01, 0.02\}$ . The number of training epochs is fixed to 1000. We found that the best hyperparameters for our models are  $\{lr = 0.1, k = 100, b = 6000, \lambda = 0.03, d = 0.2\}$ .

**Evaluation protocol.** In our experiments, we follow the evaluation protocol introduced by Zitnik et. al<sup>(8)</sup>, and we evaluate the TriVec model on the testing data split using three evaluation metrics: area under the ROC curve, area under the precision recall curve and average precision at 50 positives. The testing data contains both positive and negative data samples with a ratio of 1:1. We also use the same negative sampling procedure in the study of Zitnik et. al<sup>(8)</sup> where we only consider combination that never appear as positive to be used as negatives the results. All the reported evaluation scores in our study represent the average of its corresponding scores for all the investigated polypharmacy side-effects.

**Implementation.** We use Tensorflow framework (GPU) along with Python 3.5 to perform our experiments. All experiments were executed on a Linux machine with processor Intel(R) Core(TM) i70.4790K CPU @ 4.00GHz, 32 GB RAM, and an nVidia Titan Xp GPU. The source code and the dataset splits used in our experiments are published at: <https://github.com/samehkaaleldin/pse-kge>.

## Results and Discussion

In this section we discuss the outcomes of our experiments and we compare the predictive accuracy of our proposed approach with other state-of-the-art approaches.

**Comparison with other approaches.** (Table 2) shows the results of our experiments, where the models are compared in terms of the area under the ROC and precision recall curves and the average precision at 50. The results show that our model, the TriVec model, significantly outperforms other models with 12%, 16% and 22% margins (relative to other models’ scores) in terms of the area under the ROC, precision recall curves and with an average precision at 50 compared to the Decagon model. The results also show that our methods outperforms other knowledge graph embedding models such as the RESCAL, TransE, DistMult, ComplEx and  $KB_{LRN}$  models on all metrics.

Additionally, the results show that models such as the TransE, TriVec and ComplEx models achieve significantly high scores (above 90%) in terms of the area under both the ROC and precision recall curves. We suggest that this is due to the easy nature of the evaluation protocol that uses a 1:1 negative to positive ratio in the testing set. Therefore, we suggest that future works should adapt high negative to positive ratios such as 1:10 or 1:50.

**Other potential applications.** Knowledge graph embedding models can operate on any data in a graph form, therefore, they can be easily adopted to model different biological networks. For example, our TriVec method can be used can used for modelling drug protein target networks<sup>(49, 10)</sup> where the method is used to predict new drug protein target which can be used for drug repurposing. Our method can also be used to perform tensor factorization task on complex biological systems to model functions such as protein functions. For example, the study of Mohamed et. al.<sup>(50)</sup> shows

Table 2: Summary of the results of our experiments. † represents the results of the state-of-the-art models that are obtained from the study of Zitnik et. al. (8). \* represents the results of the state-of-the-art models that are obtained from the study of Malone et. al. (39).

Model	AUC-ROC	AUC-PR	AP@50
RESCAL † (21)	0.693	0.613	0.476
DEDICOM † (47)	0.705	0.637	0.567
DeepWalk † (48)	0.761	0.737	0.658
Concatnated Features † (8)	0.793	0.764	0.712
Decagon † (8)	0.872	0.832	0.803
TransE (20)	0.949	0.934	0.962
DistMult * (22)	0.923	0.898	0.899
KB <sub>LRN</sub> * (39)	0.899	0.878	0.857
ComplEx (23)	0.965	0.944	0.952
TriVec (This study)	<b>0.975</b>	<b>0.966</b>	<b>0.983</b>

the capabilities of the method in predicting tissue-specific protein functions where it models proteins, their biological functions and their tissues as a tensor. They then tensor factorization to predict protein function in a specific human body tissues.

**Limitations.** Our approach depends on known polypharmacy side-effects to learn efficient representations of both the drugs and the side-effects. This requires rich information about the investigated drugs *e.g.* drug interactions with other drugs and proteins. However, this type of information about the drug combinations and their associated side-effects is limited due to the recency of polypharmacy approaches and the limited clinical test applied on them (7). This problem affects the predictive accuracy of our proposed model and other graph embedding based models when they are applied on drugs with limited known assertions.

## Conclusions and Future Work

In this study, we have explored the problem of predicting polypharmacy side-effect using computer simulations. We have modelled the polypharmacy side-effects data as a knowledge graph and we have applied a knowledge graph embedding model, the TriVec model, on this knowledge to predict new polypharmacy side-effects as link prediction on knowledge graphs. We have executed experimental evaluation to compare our approach to state-of-the-art models using a standard benchmarking pipeline. The results of our experiments have shown that our approach outperforms all the state-of-the-art approaches with 10%, 13% and 22% margins in terms of the area under the ROC, precision recall curves and average precision at 50.

In our future works, we intend to investigate learning embeddings of drugs according to their chemical structure to allow learning polypharmacy side-effects of drug combinations of drugs with limited known polypharmacy assertions.

## Funding and acknowledgements

The work presented in this paper was supported by the CLARIFY project funded by European Commission under the grant number 875160, and by the Insight Centre for Data Analytics at the National University of Ireland Galway, Ireland (supported by the Science Foundation Ireland grant (12/RC/2289\_P2). The GPU card used in our experiments is granted to us by the Nvidia GPU Grant Program.

## References

1. Joanne Bowes, Andrew J Brown, Jacques Hamon, Wolfgang Jarolimek, Arun Sridhar, Gareth Waldron, and Steven Whitebread. Reducing safety-related drug attrition: the use of in vitro pharmacological profiling. *Nature Reviews Drug discovery*, 11(12):909–922, 2012.

2. Jacqueline C Bouvy, Marie L De Bruin, and Marc A Koopmanschap. Epidemiology of adverse drug reactions in europe: a review of recent observational studies. *Drug Safety*, 38(5):437–453, 2015.
3. I Ralph Edwards and Jeffrey K Aronson. Adverse drug reactions: definitions, diagnosis, and management. *The Lancet*, 356(9237):1255 – 1259, 2000.
4. Janet Sultana, Paola Cutroneo, Gianluca Trifiró, et al. Clinical and economic burden of adverse drug reactions. *Journal of Pharmacology and Pharmacotherapeutics*, 4(5):73–77, 2013.
5. Kathleen M Giacomini, Ronald M Krauss, Dan M Roden, Michel Eichelbaum, Michael R Hayden, and Yusuke Nakamura. When good drugs go bad. *Nature*, 446(7139):975–977, 2007.
6. Elizabeth D. Kantor, Colin D Rehm, Jennifer S. Haas, Andrew T. Chan, and Edward L. Giovannucci. Trends in prescription drug use among adults in the united states from 1999-2012. *JAMA*, 314 17:1818–31, 2015.
7. Nicholas P. Tatonetti, Patrick Ye, Roxana Daneshjou, and Russ B. Altman. Data-driven prediction of drug effects and interactions. *Science translational medicine*, 4 125:125ra31, 2012.
8. Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. In *Bioinformatics*, 2018.
9. Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *ICML*, volume 80 of *JMLR Workshop and Conference Proceedings*, pages 2869–2878. JMLR.org, 2018.
10. Sameh K. Mohamed and Vít Nováček. Link prediction using multi part embeddings. In *ESWC*, volume 11503 of *Lecture Notes in Computer Science*, pages 240–254. Springer, 2019.
11. James Hendler Tim Berners-Lee and Ora Lassila. The Semantic Web, a new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American*: <https://www.scientificamerican.com/article/the-semantic-web/>, May 2001. Retrieved: 2017-04-21.
12. Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Chris Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 2014.
13. Michel Dumontier, Alison Callahan, Jose Cruz-Toledo, Peter Ansell, Vincent Emonet, François Belleau, and Arnaud Droit. Bio2rdf release 3: A larger, more connected network of linked data for the life sciences. In *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014.*, pages 401–404, 2014.
14. George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, November 1995.
15. Amit Singhal. Introducing the knowledge graph: things, not strings, 2012. Google Official Blog.
16. Richard Qian. Understand your world with bing, 2013. Bing Blogs.
17. Emir Muñoz, Vít Nováček, and Pierre-Yves Vandenbussche. Using drug similarities for discovery of possible adverse reactions. In *AMIA 2016, American Medical Informatics Association Annual Symposium, Chicago, IL, USA, November 12-16, 2016*. AMIA, 2016.
18. David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. Building watson: An overview of the deepqa project. *AI Magazine*, 31(3):59–79, 2010.
19. Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.*, 29(12):2724–2743, 2017.



20. Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, pages 2787–2795, 2013.
21. Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, pages 809–816. Omnipress, 2011.
22. Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, 2015.
23. Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org, 2016.
24. Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *ESWC*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer, 2018.
25. Ni Lao and William W. Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine Learning*, 81(1):53–67, 2010.
26. Matt Gardner and Tom M. Mitchell. Efficient and expressive knowledge base completion using subgraph feature extraction. In *EMNLP*, pages 1488–1498. The Association for Computational Linguistics, 2015.
27. Sameh K. Mohamed, Vít Nováček, and Pierre-Yves Vandebussche. Knowledge base completion using distinct subgraph paths. In *SAC*, pages 1992–1999. ACM, 2018.
28. Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.
29. Rawan S. Olayan, Haitham Ashoor, and Vladimir B. Bajic. Ddr: efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. In *Bioinformatics*, 2018.
30. Yiding Lu, Yufan Guo, and Anna Korhonen. Link prediction in drug–target interactions network using similarity indices. In *BMC Bioinformatics*, 2017.
31. André C. A. Nascimento, Ricardo B. C. Prudêncio, and Ivan G. Costa. A multiple kernel learning algorithm for drug–target interaction prediction. In *BMC Bioinformatics*, 2016.
32. Nir Atias and Roded Sharan. An algorithmic framework for predicting side effects of drugs. *Journal of computational biology : a journal of computational molecular cell biology*, 18 3:207–18, 2011.
33. Wen Jun Zhang, Xiang Yue, Feng Liu, Yanlin Chen, Shikui Tu, and Xining Zhang. A unified frame of predicting side effects of drugs by using linear neighborhood similarity. In *BMC Systems Biology*, 2017.
34. Kyuho Han, Edwin E. Jeng, Gaelen T. Hess, David W. Morgens, Amy Li, and Michael C Bassik. Synergistic drug combinations for cancer identified in a crispr screen for pairwise genetic interactions. In *Nature Biotechnology*, 2017.
35. Jia Jia, Feng Zhu, Xiaohua Ma, Zhiwei W. Cao, Yixue X. Li, and Y. Z. Chen. Mechanisms of drug combinations: interaction and network perspectives. *Nature Reviews Drug Discovery*, 8:111–128, 2009.
36. Twan van Laarhoven, Sander B. Nabuurs, and Elena Marchiori. Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics*, 27 21:3036–43, 2011.
37. Shinhyuk Kim, Daeyong Jin, and Hyunju Lee. Predicting drug–target interactions using drug–drug interactions. In *PloS one*, 2013.

38. Bethany Percha, Yael Garten, and Russ B. Altman. Discovery and explanation of drug-drug interactions via text mining. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 410–21, 2011.
39. Brandon Malone, Alberto García-Durán, and Mathias Niepert. Knowledge graph completion to predict polypharmacy side effects. In *DILS*, 2018.
40. Alberto García-Durán and Mathias Niepert. Kblrn: End-to-end learning of knowledge base representations with latent, relational, and numerical features. In *UAI*, 2018.
41. Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. The sider database of drugs and side effects. *Nucleic acids research*, 44 D1:D1075–9, 2016.
42. Jörg Menche, Amitabh Sharma, Maksim Kitsak, Susan Dina Ghiassian, Marc Vidal, Joseph Loscalzo, and Albert-László Barabási. Uncovering disease-disease relationships through the incomplete interactome. *Science*, 347, 2015.
43. Andrew Chatr-aryamontri, Bobby-Joe Breitkreutz, Rose Oughtred, Lorrie Boucher, Sven Heinicke, Daici Chen, Chris Stark, Ashton Breitkreutz, Nadine Kolas, Lara O’Donnell, Teresa Reguly, Julie Nixon, Lindsay Ramage, Andrew G. Winter, Adnane Sellam, Christie S. Chang, Jodi E. Hirschman, Chandra L. Theesfeld, Jennifer M. Rust, Michael S. Livstone, Kara Dolinski, and Mike Tyers. The biogrid interaction database: 2015 update. In *Nucleic Acids Research*, 2015.
44. Damian Szklarczyk, Alberto Santos, Christian von Mering, Lars Juhl Jensen, Peer Bork, and Michael Kuhn. Stitch 5: augmenting protein–chemical interaction networks with tissue and affinity data. In *Nucleic Acids Research*, 2016.
45. Thomas Rolland and Murat Tasan et. al. A proteome-scale map of the human interactome network. *Cell*, 159:1212–1226, 2014.
46. Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, volume 9 of *JMLR Proceedings*, pages 249–256. JMLR.org, 2010.
47. Ioakeim Perros, Evangelos E. Papalexakis, Fei Wang, Richard W. Vuduc, Elizabeth Searles, Michael Thompson, and Jimeng Sun. Spartan: Scalable parafac2 for large & sparse data. In *KDD*, 2017.
48. Bryan Perozzi, Rami Al-Rfou’, and Steven Skiena. Deepwalk: online learning of social representations. In *KDD*, 2014.
49. Sameh K Mohamed, Vít Nováček, and Aayah Nounu. Discovering protein drug targets using knowledge graph embeddings. *Bioinformatics*, 08 2019.
50. Sameh K. Mohamed. Predicting tissue-specific protein functions using multi-part tensor decomposition. *Inf. Sci.*, 508:343–357, 2020.