CONSTRUCTING VECTOR REPRESENTATIONS FROM BIOLOGICAL INFORMATION TO ACHIEVE PRIVACY

CHAVALI AMARESH[1] SK YEZULLA HUSSAIN[2] SK YEZULLA HUSSAIN[3] J. RAMESH[4]
[1]ASST.PROFESSOR, DEPARTMENT OF INFORMATION TECHNOLOGY,
[2] ASST. PROFESSOR, DEPARTMENT OF INFORMATION TECHNOLOGY,
[3]ASST. PROFESSOR, DEPARTMENT OF INFORMATION TECHNOLOGY,
[4] ASST. PROFESSOR, DEPARTMENT OF INFORMATION TECHNOLOGY,
[1,2,3,4] SRI MITTAPALLI COLLEGE OF ENGINEERING

Abstract:

Despite the importance of DDI detection for patient safety, there has been very little study in this area of pharmacovigilance compared to other areas. The issue of forecasting individual pharmacological side effects has recently been tackled by using Embedding of Semantic Predictions (ESP), a technique for constructing distributed vector representations from structured biological information. Building on a previous reinterpretation of this issue as a network of drug nodes linked by side effect edges, we expand this work in the present study by applying ESP to the challenge of forecasting polypharmacy side-effects for certain medication combinations. We compare a previously published graph convolutional neural network method with ESP embeddings produced by the resultant graph on a side-effect prediction challenge, using identical data and assessment techniques. While being much simpler, more re-usable, and trained quicker, we show that ESP models perform better.

Introduction:

DDIs pose a significant threat to patient safety in the pharmaceutical industry. The practice of prescribing more than one medication to a patient at the same time is called polypharmacy. The prevalence of polypharmacy has been on the rise1, and it is notably high among some patient populations, including the elderly. Specifically, 66.8% of people aged 65 and higher used three or more prescription prescriptions between 2011 and 2014, according to the CDC[2]. While it is true that certain medications work better when taken together, many pharmacological combinations have undesirable and even dangerous physiological effects3. When two drugs operate on the same biological route, for instance, they may inhibit or change the impact of one or both drugs, increase the severity of side effects, or cause new side effects altogether due to substrate competition. It is challenging to identify and account for the consequences of polypharmacy [1-3]. It is impractical to examine every potential combination of a new medicine with every other medication on the market, much alone every exploratory agent, while conducting clinical studies to evaluate for adverse effects. Pharmacovigilance systems are in place for post-market monitoring to watch for and assist regulatory action against hazardous medication effects, because clinical studies do not capture all of a drug's side effects. The FDA's Adverse Event Reporting System (FAERS) is one example of such a system [4]. By collecting reports of Adverse Drug Events (ADEs), healthcare providers and others may look for trends in reporting, which might lead to the discovery of polypharmacy effects [5]. Several data mining techniques have been developed to identify drug/ADE associations6 as manual analysis of these reports is difficult. Relevant to the present study, Tatonetti et al. used a data mining strategy that

relied on adverse event reports5 to compile a database of DDI side effects, TWOSIDES. As the number of potential medication combinations continues to skyrocket, one of the most pressing issues in pharmacovigilance is finding computational ways to anticipate potential novel polypharmacy adverse effects. There have been many methods detailed for the purpose of anticipating medication interactions. The assumption that medications with comparable properties would engage in analogous drug interactions is the basis for several methods. Modelling based on shared chemical and structural features has been investigated for this reason [7,8]. In contrast, other methods use text mining techniques applied to large amounts of unstructured material, such as the biomedical literature database Medline.9. A third school of thought on the DDI prediction issue determines whether a certain medication combination interacts or not by using binary classification methods. One approach that shows promise is to think of DDI prediction as a link prediction problem[6–10]. In this model, medications represent nodes and side effects represent the connections between them. Unfortunately, there hasn't been a tonne of research on how to forecast DDI side effects, either in terms of their frequency or the specifics of these symptoms. A recent thorough study by Vilar, Friedman and Hripcsak[13] provides a complete explanation of previously published DDI prediction algorithms for the interested reader. Decagon11, a graph convolutional network method developed by Zitnik et al., serves as an inspiration for the present investigation. Using the idea of a graph to describe medication interactions, Decagon uses the notion of side effects as edges and pharmaceuticals as nodes. various kinds of edges reflect various sorts of side effects. Predicting the effects of many medications becomes a problem involving multiple relationships. We train graph embeddings for medications (nodes) and side effects (edges), and then we use them to forecast the kinds of connections that are likely to exist between drug combinations. Differentiating it from previous methods like Fokoue et al.'s link prediction model10 is the fact that it predicts not only the existence of an interaction but also its kind. According to reports, Decagon is the first method for simulating all kinds of polypharmacy adverse effects.the eleventh Area under the receiver operating curve (AUROC) and area under the precision-recall curve (AUPRC) measure Decagon's performance in predicting side-effects given a drug pair. Decagon is trained using data from TWOSIDES and other sources for drug target and protein-protein interaction data. When compared to many baseline multi-relational link prediction methods, Decagon achieves a mean AUROC across ADEs improvement of at least 0.1 on this task: A multi-relational tensor factorisation approach called RESCAL12 achieves an average AUROC of 0.693 by breaking down a matrix encoding drug-drug relationships into components representing drugs and side effects. Another method, DeepWalk13, uses a biassed random walk through the network's node neighbourhood to create neural embeddings, which are then used as a foundation for logistic regression. It achieves an average AUROC of 0.761. The literature also supports the use of Decagon for the prediction of new DDIs. Nevertheless, there are drawbacks to this technique as well. The first step is to train the node embeddings using a graph convolutional network14 method. This method uses d-dimensional edge matrices to connect d-dimensional node embedding vectors. Training such graph convolutional networks is computationally intensive, limiting their usefulness for bigger datasets. Unfortunately, this is an issue that will persist even as computing power grows, as the amount of data sets is growing at a comparable pace as compute power. Secondly, while the procedure generates drug vector embeddings (nodes), the side-effect embeddings (edges) are represented by matrices and are therefore difficult to reuse as they reside in a space with different dimensions than the drug vector space. Third, Decagon may not be the best option if a simpler model is more parsimonious; models with less parameters may

be better able to explain the underlying data, prevent overfitting, and generalise than Decagon. Embedding of Semantic Predictions (ESP)[16], which can handle all of these issues, is used for representation learning in our present study, which is an alternative method for DDI prediction that expands on Zitnik et al.'s graph-based understanding of the DDI prediction problem. ESP is a neural network-based technique for producing vector embeddings for medications and side effects and other biological ideas from predictions, which are triples of concepts with relationships between them. Utilising vector symbolic architectures, ESP constructs temporary embeddings for composite ideas (such INHIBITS cytochrome p-450) from their component vectors using the composition (binding) and addition (superposition) operations. Vectors representing concepts appearing in predictions with comparable predicate-argument pairings move closer to each other in vector space as a result of training-induced embedding updates. Alternatively, in vector space, vectors representing different ideas tend to be orthogonal. An n-dimensional vector space comprises 2n nearly orthogonal vectors18, and the large dimensionality of ESP vectors (e.g., 10,000 bits in binary vector implementations) makes it very improbable that vectors would be identical to each other only by accident. Prior to beginning the encoding procedure, vectors are initialised at random. Following are the modifications made to the subject (S(s)), predicate (P(p)), and object (C(o)) weight vectors for each positive triple in the training set:

The equation S(s)+= P(P) ® C(*o*) x an x (1 − NNHD(S(s), P(P) ⊎ C(*o*)) can be rewritten as: C(*o*)+= S(s) m P(P) x an x (1 − NNHD(S(s) ∅ P(P), C(*o*)) P(P)+= S(s) m C(*o*) x an x (1 − NNHD(S(s) ∅ C(*o*), P(P)) You may have noticed that ESP learns not one but two embeddings for every concept—the semantic vector (S) and the context vector (C)—instead of only one for relationships (the predicate vector P). The operators +, ∅, and ⊏ stand for binding, release, and superposition, respectively. The method of vector symbolic architecture is crucial to the execution of these operators. Binding in the Binary Spatter Code16,17 is the elementwise XOR, and we employ it here. In this version, the release and binding operators are equal since XOR is its own inverse. For each set of vectors that are "added" via the superposition operator, the element with the highest degree of elemental similarity (1 or 0) is preserved, and any ties are broken at random. The stages involved in updating are also affected by a learning rate (α) that decreases linearly and the degree of similarity (NNHD) between the intended and actual vectors [16]. For negative samples, which are generated by using -NNHD instead of 1-NNHD to replace the objects in positive triples with random ones, the update procedures are similar. The following is the definition of the cross-entropy function, which is used as the optimisation goal in stochastic gradient descent:Where (*y*, P, ¬o) is a training triple and (*y*, P, ¬o) is a training triple with a corrupted object (a negative example), the sum of the following terms is obtained: ∑(s,p,o)log(NNHD (S(s), p(p) ⊏ c(o)) + ∑(s,p,¬o)∈D'log(1−NNHD(S(s),p(p)⊗c(¬o)) +ℑ(s,p,¬o)∈D'. To rephrase, the goal of optimisation is to generate concept representations that maximise the distance between subjects and predicate-object pairs to which they do not correspond, while minimising the distance between subjects and predicate-object pairs to which they do correspond. To understand the embeddings, the ESP architecture is shown in Figure 1.

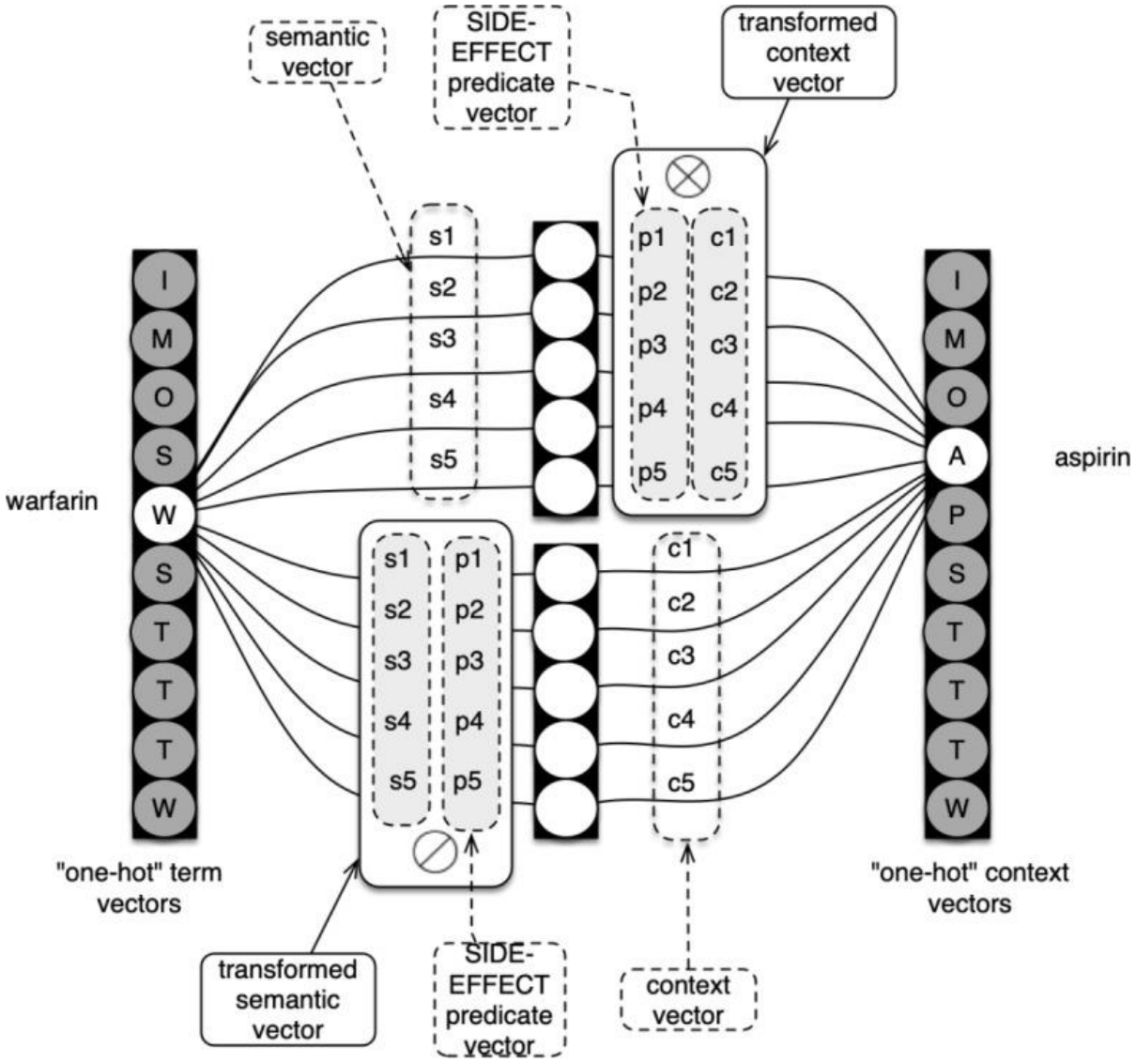

Figure 1.

A schematic of the ESP architecture that was used to produce 5-dimensional embeddings for a vocabulary of 10 concepts. Based on work by Cohen et al.16 This network uses one-hot vector representations to train weight matrices that include the vector embeddings of each drug. Each drug is activated independently during training. As an example, the input and output vectors for encoding warfarin-BLEEDING-aspirin are one-hot vectors with a 1 in the warfarin position and a 1 in the aspirin position, respectively. All columns are set to 0 except for the ones belonging to these specific medications when the vector is multiplied by the weight matrix. Hence, the encoding update phase just involves the weight matrices for a triple's ideas and predicate. The aforementioned optimisation goal is advanced with this version. We update the weight matrices in such a way that S(warfarin) ⊏ C(aspirin) moves closer to P(bleeding) and C(aspirin) moves closer to S(warfarin) ∅ P(bleeding) and P(bleeding) moves closer to S(warfarin) ∅ C(aspirin). If you're interested in learning more about the approach and how it performs on other paired entity-level tasks, we recommend reading Cohen et al.16.The reason we may search the resultant vector space for ideas linked to a particular concept is because we will learn comparable

representations for similar concepts. This allows us to compose concepts. We create drug and side effect representations using drug/side-effect/drug triples. This yields semantic vectors for the input weights, context vectors for the drugs, and predicate vectors for the side effects as output weights. If two medications' semantic and context vectors are constructed and then joined together, the resulting vector space should resemble the predicate vectors of potential polypharmacy side effects. "What polypharmacy side effects might one expect when taking warfarin and aspirin together?" is therefore an intriguing question to ask. The response is the predicate vector that minimises the product of all potential adverse effects of S(warfarin) and C(aspirin), taking into account all conceivable interactions between the two drugs. ESP has a track record of efficacy when it comes to forecasting pharmacological side effects16. In a recent study, Mower et al.[15] built an ESP model using concept-relationship-concept triples from SemMedDB, a knowledge base extracted from the MEDLINE corpus of medical literature using SemRep[14]. This model then fed into a supervised machine learning model that was trained to predict which drugs might have side effects. The model achieved strong results, with an area under the receiver operating curve (AUROC) of 0.96 and 0.95 among two test sets. The issue of whether ESP may be used to anticipate DDIs as well arises from the effectiveness of the method in forecasting specific pharmacological side effects. We investigate the possibility that ESP might be helpful for the polypharmacy side effect prediction problem in this article.

Based on an analysis of 963 side effects, Table 2 displays the performance metrics for ESP and Decagon. The 8-epoch ESP model has an average AUROC of 0.903 (range: 0.841-0.977), an average AUPRC of 0.875 (range: 0.779-0.976), and an average AP@50 of 0.865 (range: 1.01-1.0). With an average AUROC of 0.855 (range 0.259-0.949), an average AUPRC of 0.793 (range 0.366-0.934), and an average AP@50 of 0.638 (range 0.066-0.950), these findings more than outperform both the Decagon results reported in Table 2 and our locally-trained Decagon model after 8 epochs. Including setup and prediction time on the test set, our 4-epoch ESP model takes about 3.5 hours to train with 16,000 dimensions (bits). About fifty minutes elapsed between each period. Decagon's four training epochs took an average of 36 hours each, while the whole process took six days and four hours (including setup and test set prediction). Figure 2 shows the average area under the receiver operating characteristic curve for 963 side effects with varying vector dimensionalities, which were generated by truncating the 16,000-dimensional vectors.

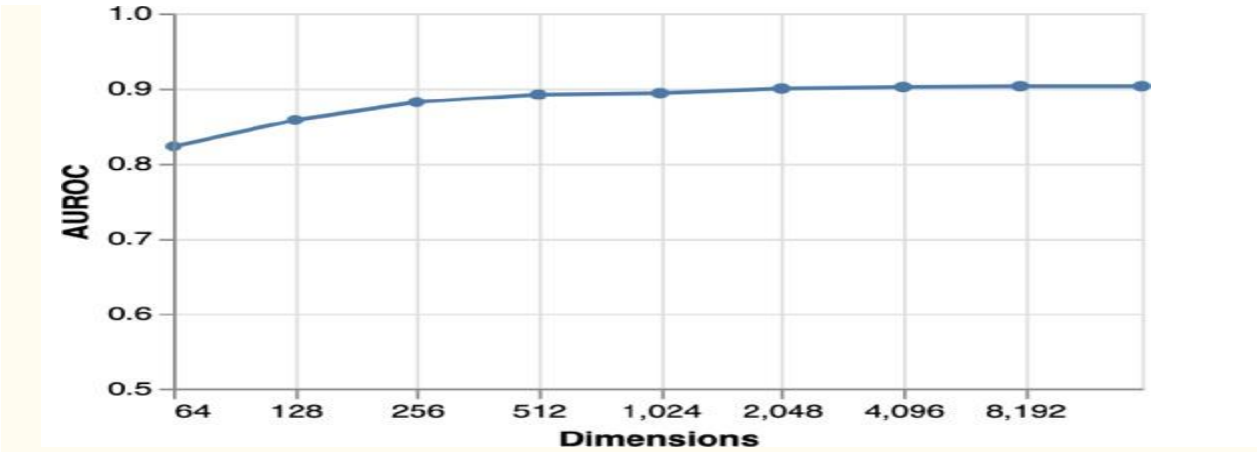

Figure 2.

Our locally-trained model and the best reported performance both show that ESP is superior than Decagon in predicting the adverse effects of polypharmacy. On top of that, ESP trains for 8 epochs at a rate that is 43.2 times quicker than Decagon, but it still manages to reach the same performance. Unfortunately, the exact number of epochs until early halting was not recorded, however the top performing Decagon models were run up to 100 epochs. Due to resource limitations, we could only train our locally-trained model for a maximum of 8 epochs. This explains why there is a little performance discrepancy with the findings that have been published before. We also point out that ESP generates embeddings of the same dimensions for medications and side effects that can be reused, which may be applied to other issues by means of vector symbolic architecture procedures. We successfully deciphered the model's embeddings by using this method in conjunction with UMAP projection visualisation.
It is worth noting the impressive achievement that ESP managed to attain. The enhanced performance of ESP over Decagon places this model among the best in its class, especially when one considers that Decagon significantly exceeded other baseline techniques. Decagon has an advantage over other polypharmacy side effect prediction models that ESP shares: it can represent and model multiple side effects simultaneously using a multimodal network. This allows for the sharing of model parameters across side effects, which in turn improves performance. The model can learn to generalise across similar concepts, like "kidney failure" and "acute kidney failure," as shown in Table 1. A quicker model is better if there is no need to sacrifice accuracy for its speed, even if training time is not the most important element when there are enough of computing resources to choose from. Scientific advancement via greater equity is made possible by reducing the substantial hardware requirements. This allows university researchers with very modest computer resources to replicate and expand upon existing studies. Training complex neural network models can consume a lot of energy and produce a carbon footprint that is five times greater than an average car's, including manufacture30. This is true even for research groups that have access to powerful computing clusters and numerous GPUs. When compared, common laptops or servers may teach ESP. While it's true that computing power is increasing at a rate consistent with Moore's law, many believe that resource-intensive computations will become easier over time. However, this is a hard argument to make in the field of informatics, where data sources are growing in size at an incredible rate.24. Biomedical literature, EHR data, social media, and adverse event reports are all being used more and more, and their development is being tracked, sometimes at an exponential rate (as with EHR and patient reported data). At present, it is recommended to retrain DDI prediction models whenever new data on adverse events becomes available, which might be on a weekly or monthly basis. Training costs will rise even higher as the availability of massive amounts of patient-level data opens the door to the possibility of tailoring DDI prediction models to each patient's unique demographics, health state, and genetic composition. By enabling the use of commodity hardware, ESP may aid in promoting research equality and corresponds to previously articulated suggestions to prioritise the creation of efficient models. One benefit of ESP16 is its capacity to develop representations that match how humans see similarities between ideas. Table 3 shows that our study successfully shown that learnt embeddings for related side effects, including acute renal failure and kidney failure, are identical. This search does, however, also provide results for other forms of organ failure. The fact that FAERS includes individuals with multiple organ failure suggests that various forms of organ failure may start to cluster together. By associating these organ failures with reports that include the same medication pairs—for example, pharmaceuticals used in sepsis—a statistical model

may identify connections between these drugs and all types of organ failure. Using the bound product S(aspirin)⊎ C(warfarin) in a similarity-based search for side effects, hemarthrosis, parotitis, and alveolitis were among the results. Warfarin is known to cause swelling and discomfort in the salivary glands, while aspirin has a history of being linked to the development of lung illness. No evidence has been found to suggest that aspirin or warfarin may worsen these side effects. Since both warfarin and aspirin are anticoagulants, it is not surprising that we discovered a resemblance between S(aspirin)⊎ C(warfarin) and hemarthrosis, a bleeding illness. Please be aware that indicators, not side effects, may be referenced in similarity search results. Tatonetti et al.5 attempted to ensure that indications would not be included as side effects in the TWOSIDES set by applying corrections for confounding. However, it seems that this effect was not fully eliminated, so our model is still learning from indication-related statistical patterns in this dataset. Our projection visualisation could not successfully group all side effect representations for a certain side effect category. For instance, although symptoms affecting the neurological system are distributed over the vector space, those affecting the reproductive system, cognitive disorders, and haematopoietic systems are neatly contained inside their own classes. We attribute this in part to the small variety of data types used to construct the training set. Polypharmacy side effects and medication target information are the two pieces of data that ESP might potentially incorporate into the concept embeddings. An further layer of complexity is the fact that, as mentioned earlier, side effects tend to cluster since they are really indications shared by certain medication combinations. In their discussion of this topic, Zitnik et al.11 noted that several side effects often occur together in this dataset. Consequently, for more complex data sets with more kinds of concepts and relations, which might lead to more significant similarities among the learnt side effects, co-location of embeddings in space is not as interpretable. Similarities in prior ESP applications have been easier to understand since they used extensive data sets with many relationship kinds and notions. The ESP model's simplicity is one of its best qualities. Decagon uses 16,000-dimensional vectors with around 87 million trainable parameters, each of which is 32 bits in size, for a total of about 2.8 billion bits of training parameters; ESP, on the other hand, gets better results with just about 36 million single-bit parameters, or more than seventy-seven times less representational capacity. The fact that ESP approaches Decagon performance even with 1,024 dimensions is remarkable. This dimensionality equates to a mere 2 million trainable bits of representational power, which is more than three orders of magnitude lower than Decagon. With 64 bits per embedding, ESP outperforms the baseline techniques described by Zitnik et al., even though there are only 144,000 bits of trainable parameters. It follows that predicting DDIs in TWOSIDES does not need the massive representational capacity of a trained Decagon model. To uncover any performance benefits that its bigger capacity and more computationally intensive training technique might provide, additional rigorous testing may be necessary. Several limitations are present in this study. An essential caveat is that training data could not be a perfect match for ground reality. Data retrieved from text sources is housed in TWOSIDES. There are a lot of unvalidated associations in the database, including some strange side effects like "Mumps" or "Fracture," which might not be able to pass curation. This is in addition to the few novel drug interactions that have been confirmed in TWOSIDES through research of pertinent patient records and laboratory experiments. We may learn more about the potential medication interactions from these side effects. For example, if a specific combination of pharmaceuticals is associated with an upsurge in cases of mumps, it would be reasonable to assume that the polypharmacy side effect is a reduction in immune system function. Similarly, if a community of patients using a certain combination of medications has an

uptick in fractures, it's reasonable to assume that this is due to a reduction in bone density brought on by the interactions between the medicines. However, data mining is not as good as using a curated reference collection of medication interaction data. We choose to utilise the same datasets as a similar prior technique since, as mentioned before, it is difficult to make models directly comparable. The lack of encoding for the directionality of links between ideas is another drawback of this technique. One way to look of DDIs is as symmetrical drug relationships; after all, if drug A interacts with drug B, then drug B must likewise interact with drug A. Following the work of Zitnik et al., we create a new sort of side effect called the "reverse relationship" by inverting and appending all DDI triples in the dataset during preprocessing. Put simply, for every edge (with an edge type related to side effects) that connects medication A to drug B, we also include an edge (with a different edge type) that connects drug B to drug A. Drug embeddings gain from the extra triples (edges) during training, but the predicate embeddings—representations of redundant side effects—that they provide are not used for prediction purposes. However, the maintained side effect embedding does not encode the inverse connection, even if it is known to be true. For drug targets, the process involves inverting the triple and adding a new edge type to the dataset. This is in line with the directionality of the connection in the drug target case: we cannot state that protein B targets drug A only because drug A targets protein B. Incorporating drug target triples with the same edge type in reverse would make directionality difficult to discern. The last group to be addressed is protein-protein interaction (PPI) triples, which are fundamentally symmetric. By inverting the order and then establishing an alternate relationship, Decagon quadruples all PPI triples. This results in a total of four edges connecting the nodes for protein A and protein B. These edges are two "interacts_with" edges, one pointing from A to B and one from B to A, and two "reverse_interacts_with" edges, one pointing from A to B and one from B to A. Although ESP is capable of directly encoding directionality, we opted not to investigate this in this study. Instead, we used the same technique for preparing the input data as Zitnik et al. to ensure direct comparison.

Conclusion:

Many individuals are at risk of experiencing adverse events caused by drug-drug interactions. Because DDIs are poorly characterized before drugs are released to market, efficient and accurate analysis of post-market surveillance data is critical to patient safety. ESP, a representation learning method with demonstrated utility in the pharmacovigilance space, provides an alternative approach to DDI prediction. In this work, we have shown that application of ESP to adverse drug event data can yield accurate predictions of polypharmacy side effects, with performance exceeding that of the best reported machine learning models on this task. Being considerably more lightweight in terms of required compute power and model complexity than previously published approaches, ESP can be expected to scale up without imposing undue financial and ethical burdens on researchers, particularly as the volume of data, number of models to train, and frequency of re-training increase. Additionally, we have demonstrated that ESP produces meaningful representations of biomedical concepts that can enable biomedical discovery in novel ways, providing insights that pave the way for further development of ESP models for pharmacovigilance and other applications.

References:

1. Percha B, Altman RB. Informatics confronts drug–drug interactions. *Trends Pharmacol Sci.* 2013 Mar;34(3):178–84. [PMC free article] [PubMed] [Google Scholar]

2. National Center for Health Statistics. Health, United States. 2017;2017 [Google Scholar]

3. Vilar S, Friedman C, Hripcsak G. Detection of drug-drug interactions through data mining studies using clinical sources, scientific literature and social media. *Brief Bioinform.* 2018;19(5):863–77. [PMC free article] [PubMed] [Google Scholar]

4. U.S. Food Drug Administration. FDA Adverse Event Reporting System (FAERS). [Internet]. Available from: https://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/ ucm070093.htm. [Google Scholar]

5. Tatonetti NP, Ye PP, Daneshjou R, Altman RB. Data-driven prediction of drug effects and interactions. *Sci Transl Med.* 2012 Mar 14;4(125) [PMC free article] [PubMed] [Google Scholar]

6. Kastrin A, Ferk P, Leskoek B. Predicting potential drug-drug interactions on topological and semantic similarity features using statistical learning. *PLoS One.* 2018;13(5):1–23. [PMC free article] [PubMed] [Google Scholar]

7. Takeda T, Hao M, Cheng T, Bryant SH, Wang Y. Predicting drug-drug interactions through drug structural similarities and interaction networks incorporating pharmacokinetics and pharmacodynamics knowledge. *J Cheminform.* 2017;9(1):1–9. [PMC free article] [PubMed] [Google Scholar]

8. Zhang P, Wang F, Hu J, Sorrentino R. Label propagation prediction of drug-drug interactions based on clinical side effects. *Sci Rep.* 2015 Dec 21;5(1):12339. [PMC free article] [PubMed] [Google Scholar]

9. Percha B, Garten Y, Altman RB. Discovery and explanation of drug-drug interactions via text mining. *Pac Symp Biocomput.* 2012;73(11):410–21. [PMC free article] [PubMed] [Google Scholar]

10. Fokoue A, Sadoghi M, Hassanzadeh O, Zhang P. Predicting drug-drug interactions through large-scale similarity-based link prediction. *In: International Semantic Web Conference.* 2016:774–89. Springer. [Google Scholar]

11. Zitnik M, Agrawal M, Leskovec J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics.* 2018;34(13):457–66. [PMC free article] [PubMed] [Google Scholar]

12. Nickel M, Tresp V, Kriegel H. A three-way model for collective learning on multi-relational data. *In: ICML.* 2011:809–16. [Google Scholar]

13. Perozzi B, Al-Rfou R, Skiena S. DeepWalk. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*. 2014:701–10. New York, New York, USA: ACM Press. [Google Scholar]

14. lmcinnes/umap: Uniform Manifold Approximation and Projection. [Internet]. GitHub; [cited 2019 Mar 11]. Available from: https://github.com/lmcinnes/umap/

15. Anusha, Pureti, T. Sunitha, and Mastan Rao Kale. "Detecting and Analyzing Emotions using Text stream messages." *ECS Transactions* 107.1 (2022): 16913.

16.Aharonu, Mattakoyya, et al. "Entity linking based graph models for Wikipedia relationships." *Int. J. Eng. Trends Technol* 18.8 (2014): 380-385.