CSE 6250 Final Project

Anonymous submission

Abstract

CSE 6250 Project Repo
Project Video
4SDrug Repo
SafeDrug Repo

Introduction

The original paper 4SDrug explores the realm of drug recommendations and offers an approach that exceeded other models at the time. Through research of related work, the paper takes note of missed opportunities and concerns for patient health, finance, and privacy. Notable points raised:

- Patient health and finances are often put at risk due to lack of considerations for the number of drugs and drug-to-drug interactions
- Usage of deep learning and unordered sets had not been explored fully yet due to graph learning popularity

The paper then starts to dive into its framework on how it tackles the important aspects missing from the related works. It dives deep into each aspect of the model on how it accounts for set-to-set comparison, weights on symptoms to signify higher or lower importance, and constraints/penalties to enforce the model to recommend a lower number of drugs and drug-to-drug interactions. Our project aims to reproduce the results.

Scope of Reproducibility

The research questions we plan to reproduce (RQs):

1. How does 4SDrug perform in comparison to state-of-the-art recommendation methods?
2. How do the hyperparameters affect the recommendation performance and how to choose optimal values? (Tan et al. 2022)

Due to data availability, we will only look to replicate results using MIMIC-III data and the code provided to us.

Methodology

Dataset Description

The data for this paper comes from the MIMIC-III dataset (Johnson et al. 2016). These data are a collection of deidentified electronic health records (EHRs) of over forty-thousand patients from the Beth Israel Deaconess Medical Center in Boston, MA between 2001 and 2012.

Items	4SDrug	Reproduce
# of visits	27,869	15,032
# of symptoms	1,113	1,958
# of drugs	131	112
avg # of symptoms per symptom set	31.81	13.63
avg # of drugs per drug set	14.36	19.57
total # of DDI pairs	448	674

Table 1: Data Statistics - Recreating 4SDrug Table 2

The 4SDrug paper utilizes work done by the SafeDrug paper (Yang et al. 2021) for its data collection and processing. This is in part because SafeDrug is used as a benchmark to compare results, and the analysis is more meaningful by starting with the same data. Since our goal was to reproduce the 4SDrug paper and not SafeDrug, we looked at data processing at a high level only.

The SafeDrug paper uses the **PRESCRIPTIONS**, **DIAGNOSES_ICD**, and **PROCEDURES_ICD** tables from the MIMIC-III dataset to create tensors of patient visits. Each tensor represents a patient and contains tensors of visits. Each visit contains three tensors with the set of diagnoses, procedures, and prescriptions associated with that visit. After the preprocessing done by SafeDrug, this dataset includes 6,350 patient records and is stored as "records_final.pkl" in the "Output" folder of the SafeDrug repo. We used this file as our starting point.

Before use in 4SDrug, these patient records are "flattened" to remove the patient dimension so that all visits are independent of each other. This flattened tensor contains 15,032 visit records. Finally, these records are split into training, testing, and evaluation sets using a 4:1:1 split. The paper describes the size of the split, but both the paper and code repo are unclear on how the split was made. For our purposes, we split the data consecutively using the first 10,020 visits as training data.

In addition to providing patient record data, SafeDrug also provides a data set of harmful drug-to-drug interactions (DDIs). These data are important for helping the 4SDrug algorithm not prescribe drugs that include harmful interactions.

Table 1 compares data summary statistics found in 4SDrug Table 2 to the data we were able to recreate from SafeDrug. As can be seen, the visit data we have is quite different from what is reported is 4SDrug. Interestingly, although we have about half as many visits, the number of unique symptoms is larger by about 800. Despite these differences, we were able to get similar model results as described later.

Model Description

The first step in modeling 4SDrug recommendation is to create an embedding for each set of symptoms and prescribed drugs. This embedding is similar to word embeddings in natural language processing (NLP) and it helps standardize set lengths (Equation 1 in 4SDrug). Symptoms are weighted by importance in the embedding based on a learned parameter (Equation 6 in 4SDrug). The embedding has a default size of 64, but this parameter can be changed by the user.

Ultimately, the goal of the model is to recommend a set of drugs based on a set of symptoms. To begin this process, individual drugs are compared to each set of symptoms. Equation 1 (Equation 3 in 4SDrug) shows this calculation as the element-wise dot product of a set of symptoms ($h_S^{(i)}$) to a single drug (d_j) through a sigmoid (σ) function. This function returns the probability that drug j should be included in the recommendation for symptom set i. While not set up like a traditional neural network (NN) model, this calculation can be compared to a hidden layer in a NN.

$$g\{h_S^{(i)}, d_j\} = \sigma(h_S^{(i)} \odot d_j) \tag{1}$$

The model uses a multi-label binary classification loss function (Equation 5 in 4SDrug) so recommended drugs have probabilities approaching 1. Other loss functions are described in detail in the Training section.

The learned parameters of the models are vectors that convert symptoms and drug sets into their embeddings, and the importance weights on the symptoms. Using 4SDrug's and our reproduced values for number of symptoms and drugs and an embedding size of 64 we get the following calculations of the total number of parameters:

$$(1,113 * 64) + (131 * 64) + (1,113) = 80,729$$

$$(1,958 * 64) + (112 * 64) + (1,958) = 134,4338$$

The code for training the 4SDrug model can be found at the GitHub repository linked in this paper's abstract section.

Training

Computational Implementation

To run the 4SDrug model, we set up the code base on Google Colab and used a T4 GPU. Using this runtime environment and the default arguments (batch_size = 50 and epochs = 200) the code takes 32.5 minutes to run, about 10 seconds per epoch.

Loss Functions

As mentioned in the Model Description section, the 4SDrug model uses a multi-label binary classification loss function to optimize model parameters. Equation 2 shows this loss function, where $h_S^{(i)}$ and $\mathcal{D}^{(i)}$ represent the embedding symptom set and drug set for record i respectively, and function g as defined in Equation 1 (Equation 5 in 4SDrug).

$$\mathcal{L}_{rec}^{i} = \sum_{d_j \in \mathcal{D}^{(i)}} \log g\{h_S^{(i)}, d_j\} + \sum_{d_j \in (\mathcal{D} - \mathcal{D}^{(i)})} \log\Big(1 - g\{h_S^{(i)}, d_j\}\Big) \tag{2}$$

In addition to the binary classification loss function mentioned in the previous section, the model implements two other loss functions to assist the recommendations to be small and safe.

Before calculating the "small" loss function, a new set is defined (see Section 3.4.1 in 4SDrug). The authors of 4SDrug proposed that importance of drugs can be learned by intersecting symptom sets with similar symptoms ($h_{S_\cap}^{(i,\mathcal{N}_i)}$) and intersecting their prescribed drug sets ($\mathcal{D}_\cap^{(i,\mathcal{N}_i)}$). The "small" loss function is identical to the binary classification loss function except it uses these intersected sets, as shown in Equation 3 (Equation 10 in 4SDrug).

$$\mathcal{L}_{inter}^{i} = \sum_{d_j \in \mathcal{D}_\cap^{(i,\mathcal{N}_i)}} \log g(h_{S_\cap}^{(i,\mathcal{N}_i)}, d_j) + \sum_{d_j \in (\mathcal{D} - \mathcal{D}_\cap^{(i,\mathcal{N}_i)})} \log\Big(1 - g(h_{S_\cap}^{(i,\mathcal{N}_i)}, d_j)\Big) \tag{3}$$

The loss function for considering "safe" drug sets is made up of two loss functions, one with a knowledge-based penalty and one with a learned penalty. For MIMIC-III data, a drug knowledge base (DKB) is utilized, specifically TWOSIDES ((Tatonetti et al. 2012)), which provides a score A_{kl}^{d}. If drugs d_k and d_l have a DDI, then the score is 1. This is utilized in the following loss function (Equation 11 in 4SDrug):

$$\mathcal{L}_{K-DDI}^{i} = \sum_{d_k \in \mathcal{D}} \sum_{d_l \in \mathcal{D}} A_{kl}^{d} \cdot g\{h_S^{(i)}, d_k\} \cdot g\{h_S^{(i)}, d_l\} \tag{4}$$

Lastly, a data-driven DDI penalty is calculated (see Section 3.4.2 in 4SDrug). This loss function penalizes drugs that appear infrequently together in drug sets and promotes those that appear together often. Drug sets similar to those found in Equation 3 are calculated and used in the following loss function to create this penalty (Equation 13 in 4SDrug):

$$\mathcal{L}_{D-DDI}^{i} = \sum_{d_k \in \mathcal{D}_-^{i}} \sum_{d_l \in \mathcal{D}_-^{(\mathcal{N}_i)}} g\{h_S^{(i,\mathcal{N}_i)}, d_k\} \cdot g\{h_S^{(i,\mathcal{N}_i)}, d_l\} \tag{5}$$

The final loss function is a linear combination of the loss functions described above with tunable parameters α and β to control the effects of "small" and "safe" on the model.

$$\mathcal{L} = \mathcal{L}_{rec} + \alpha\mathcal{L}_{inter} + \beta(\mathcal{L}_{K-DDI} + \mathcal{L}_{D-DDI}) \quad (6)$$

Training Method

The 4SDrug model follows a single hidden-layer neural network framework. Parameter values are set at random initially and similarity scores are calculated on a forward pass. Then, parameter values are updated using gradient descent on a backward pass. Each forward/backward pass makes up an epoch.

LLM Assistance

In preparing the Methodology and Training sections, ChatGPT was given the 4SDrug paper and asked, "Can you explain Section 3 in detail to me? Specifically, how the model converts sets of symptoms and drugs to learned model parameters on drug recommendations." The full response can be found on the ChatGPT link. I found the response detailed yet easy to understand. It summarized information section by section which made the process easier to follow. In comparing the LLM output and the 4SDrug paper I could not find any informational differences.

After providing a summary, ChatGPT asked if I wanted a diagram representing the process. I agreed, and unfortunately, the diagram it created was cropped on the left so it only showed the last few steps. The chat can be found at this link.

Evaluation

Results

Reproducing Experimental Results

Using our data as defined in Section Dataset Description, we ran 4SDrug using the default model parameters as described in the paper and the GitHub repository. Table 2 shows the results of 4SDrug from the paper compared to our results.

The Jaccard score is a measure of how similar two sets are where a score of 1 represents being identical. Here, Jaccard is used to measure predicted drug sets compared to ground truth. As can be seen, our data produced worse results than the 4SDrug paper. Based on other metrics, it appears that our training of the model is recommending too few drugs as compared to actual recommendations.

When compared with other state-of-the-art methods, our training of 4SDrug does not perform as well as other methods, namely SafeDrug and GAMENet (Shang et al. 2019). Without starting with the correct data, it is difficult to ascertain if our differences are due to data differences or our implementation of 4SDrug.

Hyperparameter Experiment

α is a weight that affects the intersection loss function that is meant to guide the model to a smaller number of suggested drugs. The intersection loss function works by looking at a specific symptom set, the model will then rank all other symptom sets by the largest Jaccard coefficient. Then for these two matched symptom sets, we compare the respective drug sets and find the intersection. The model will then aim to maximize for intersecting drugs and minimize for the set of drugs outside the intersection. As a result the model is taught to keep the drugs that have worked for two symptom sets matched by the highest Jaccard coefficient.

β is a weight that affects the safe drug set principle which is made up of two components. The first applies a drug knowledge base (DKB) in the form of a matrix, if the two drugs compared are in the matrix then the output of the matrix will be one and a penalty will be applied. Since the model wants to minimize the output, it will learn to avoid suggesting these pairs of drugs. The second component is utilized when the DKB is not applicable, this component uses the intersection logic previously discussed and takes the complement each compared drug set to verify how often two drugs are not shared in the same drug set given the intersecting symptom sets. Finally these two components are summed and weighted by β.

Other parameters were not experimented with since the amount of computing resources available were limited. By conducting a grid experiment with different values for the two weights α in $\{0.1, 0.5, 1.0\}$ and β in $\{0.5, 1.0, 1.5\}$ totaled to 9 experiments. Our results proved to align with the paper's results. A higher α led to a smaller average number of medications recommended across all three β experiments. By increasing the β and holding α constant, the DDI and number of medications would increase, while the Jaccard coefficient with the ground truth would drop.

As seen in Table 3, the best results exist when $\alpha \in \{0.5, 1.0\}$ and $\beta \in \{0.5, 1.5\}$. Due to resource constraints, fine-tuning any further was not possible. It is worth noting that the impact of β was higher than α since it impacted both the knowledge graph and non-intersecting drug sets for DDI decision making.

Discussion

Data Reproducibility

After working on reproducing the results of the 4SDrug paper, we have decided that given the paper and the code repository, this work is not reproducible.

The first challenge we ran into was the data. The paper describes using MIMIC-III data and extracting symptoms from clinical notes (see Section 5.1.1 in 4SDrug). The 4SDrug GitHub repository only offers the following data guide, "Please download the MIMIC-III dataset" (Melinda315 2022).

A closer look at the code (specifically "utils/dataset.py") shows that the code expects the files "ddi_A_final.pkl", "voc_final.pkl", and "data_X.pkl" where X is "train", "eval" and "test". Further digging led to understanding that the "ddi" and "voc" files come directly from the SafeDrug repo (ycq091044 2021), and that the "data" files are calculated from the "records_final.pkl" file from SafeDrug.

In total, 4SDrug could have done a better job in allowing their data to be reproducible. Firstly, they could have included the files mentioned above in their repository. In its

Method	Jaccard	F1	Avg # of Drug	DDI Rate	$\|\Delta\|$Avg # Drug	Δ% DDI Rate
4SDrug (paper)	0.5041	0.6581	17.5040	0.0600	3.1440	-26.83%
4SDrug (reproduce)	0.4396	0.6024	13.2414	0.0525	0.856	-38.24%

Table 2: Experimental Results - Recreating 4SDrug Table 3

α	β	**best_ja**	**avg_med**	**DDI_rate**
0.1	1.5	0.4595	14.9808	0.0477
0.1	1.0	0.4636	15.2901	0.0548
0.1	0.5	0.4705	15.8228	0.0641
0.5	1.5	0.4537	14.3707	0.0479
0.5	1.0	0.4601	15.1269	0.0555
0.5	0.5	0.4646	15.3081	0.0632
1.0	1.5	0.4514	14.2442	0.0497
1.0	1.0	0.4556	14.9358	0.0561
1.0	0.5	0.4592	15.3057	0.0631

Table 3: Impact of α and β on best Jaccard score, average medications, and DDI rate.

current state, the repository is not self-sufficient and users must find the SafeDrug data on their own.

However, as shown in this article, acquiring the SafeDrug data still does not reproduce the results of the 4SDrug article. This may be due to the fact that SafeDrug pushed updates to their repository after 4SDrug was published. Tests with different versions of the SafeDrug "records_final.pkl" data did not yield any better results than those described in this paper.

Tuning and Experimenting Reproducibility

While reproducing the experimentation for hyperparameters, we were limited by the computing resources available. Through Google Colab we could complete training for one model with 100 epochs and .001 learning rate in about 15 minutes. After running 4 experiments, the T4 GPU runtime type would timeout since the limit for free usage had been reached. Using the CPU to run it would work at about 1% of the speed of the GPU and was not a viable option. Therefore we were limited to the amount of parameters and level of fine tuning due to computing resources.

Overall Comments

This paper makes a key assumption that the drugs recommended as recorded in the data are effective at treating the set of symptoms. In real life, drugs can be prescribed that have no effect or an adverse effect on treating the patient's symptoms. We could not find discussion of this assumption in the paper and thus the model may be adversely impacted by ineffective drug recommendations.

Author's Contributions

The following sections were written by Devin Warner:

- Methodology
- Training
- Discussion:Data Reproducibility
- Results:Reproducing Experimental Results

Devin worked on data preprocessing and attempting to reproduce results from Table 2, Table 3, and RQ1 from the 4SDrug paper.

The following sections were written by Justin Suen:

- Introduction
- Scope of Reproducibility
- Results: Hyperparameter Experiment
- Discussion: Tuning and Experimenting Reproducibility
- Reproducing the results from RQ3

References

Johnson, A. E. W.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. Accessed: 2025-04-08.

Melinda315. 2022. 4SDrug: Symptom-based Set-to-set Small and Safe Drug Recommendation. https://github.com/Melinda315/4SDrug. Accessed: 2025-04-08.

Shang, J.; Xiao, C.; Ma, T.; Li, H.; and Sun, J. 2019. GAMENet: Graph Augmented Memory Networks for Recommending Medication Combination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1126–1133. AAAI Press.

Tan, Y.; Kong, C.; Yu, L.; Li, P.; Chen, C.; Zheng, X.; Hertzberg, V. S.; and Yang, C. 2022. 4SDrug: Symptom-based Set-to-set Small and Safe Drug Recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, 3970–3980. New York, NY, USA: Association for Computing Machinery. ISBN 9781450393850.

Tatonetti, N. P.; Ye, P. P.; Daneshjou, R.; and Altman, R. B. 2012. Data-driven prediction of drug effects and interactions. *Science Translational Medicine*, 4(125): 125ra31–125ra31.

Yang, C.; Xiao, C.; Ma, F.; Glass, L.; and Sun, J. 2021. SafeDrug: Dual Molecular Graph Encoders for Safe Drug Recommendations. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*.

ycq091044. 2021. SafeDrug: A Safe Drug Recommendation Framework for Electronic Health Records. GitHub repository. Available at: https://github.com/ycq091044/SafeDrug, Accessed: 2025-04-08.