## Link to Code

#### Google collab notebook :

#### Github Link : Power point link :

#### Video Link:

#### Instructions to run

You will need to make a shortcut in your own drive and change the root\_path on the top of the notebook.

## Introduction

Gamenet GAMENet: Graph Augmented MEmory networks for recommending medication combination by tackles the limitation of instance-based recommendations with patient history to give better recommendations. It uses previous work that uses memory banks augmented by DDI graphs and dynamic memory based on patient history.

It uses a novel deep learning approach by constructing 2 RNN hidden layers, one for diagnosis and one for procedure. It uses this constructed patient representation with a pre-generated memory bank from DDI graph and EHR graph. The papers claim that this approach outperforms all current baselines in effectiveness measures, and achieved 3.60% DDI rate reduction. (Shang, J. et al (2019))

## Scope of reproducibility:

Hypothesis :

The papers claim that this approach outperforms all current baselines in effectiveness measures, and achieved 3.60% DDI rate reduction. We will reproduce the results with our model to verify this measure is achieved.

## Data

* The data raw data is the diagnoses, prescriptions, procedure data from MIMIC-3 . A freely-available database comprising deidentified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012.( Johnson et al , 2016)
* The data also includes drug-drug interaction (TWOSIDEs dataset) .Drug side effects and drug-drug interactions were mined from publicly available dat. (Offside , n. d.)

## Statistics on data

| Name | Statistic |
| --- | --- |
| Number of patients considered | 6350 |
| Total Clinical events | 15016 |
| Total Diagnosis ( | 1958 |
| Total Medications prescribed | 145 |
| Total Procedure | 1426 |
| Avg Diagnoses per patient | 10.514717634523175 |
| Avg medicines per patient | 8.80420884389984 |
| Avg procedures per patient | 3.8445657964837507 |
| Avg visits per patient | 2.3647244094488187 |

## Refining the dataset

#### Data Split

Training data = 8/10

Validation data = 2/10

#### Create patient visit feature DB

* we will remove duplicate entries from the 3 datasets
* the values will be sorted by subject\_id
* there will be one row for each visit in the input dataset to the model
* we will group all the diagnoses, medication, and procedure codes into arrays for every visit. The medication array would be one of the expected outputs
* we will convert NDC to ATC
* we might do some limits on the number of procedures, medications, and diagnoses to keep in our testing due to processing constraints

#### Creating feature vocabularies and patient record

Each diagnosis code, Medication NDC code, and procedure code will be mapped to a unique feature value. These feature vocabularies will be used in our modeling

We will use the unique value to create another state of patient record which is numerical and therefore can be processed by the GAMENET model and any other model we may compare it against.

#### Creating adjacency matrix

The adjacency matrix of DDI and NDC codes is needed to create the “memory banks” by running them through GCNs. These will be convoluted with input patient presentations to give an output. This is only needed to validate the results proposed in the model.

# Model

Our final model will follow the architecture outlined in the paper.

A diagram of a diagram

Description automatically generated

#### Data Input

There is 2 visit sequences that are ran through the model the include history of procedures and diagnoses by another patient . They are of size , (visit\_dim, procedure\_count) and (visit\_dim,diagnoses\_count). Visit size will include the 0th visit to the current visit. We will also send the in the medication in the same format as it will be used for key value store.

For initial experiment we are using

| Diagnosis count | 1958 |
| --- | --- |
| Medications prescribed count | 145 |
| Procedure count | 1426 |

#### Embedding Network

Each vector(medication, diagnosis, procedure) will be reduced using an embedding which will output a size emb\_dim which is set at 64 for now.

#### Dual RNN

GRU will be used for this step . We will have one GRU for diagnoses and one for procedures. The input dimension would be 64 as that is the output of embedding network. The output dimension will be doubled.

#### Patient Representation and query

We combine the GRU results by doing a concatenation to create a patient representation of procedure and diagnoses.

We run this through a linear layer with RelU to get a queries for the entire sequence.

The current query is the last element .

#### Memory Bank

This is created using the adjacency matrix created in the data pre-processing sections. We created 2 Graph convoluted networks, one for EHR adjacency matrix and one for DDI. We use the implementation provided in the paper to create these GCNs. It doesn’t have input, just weights associated with the adjacency matrix that are adjusted during training

#### Key Value store

The keys are all the queries for the past visits. This is taken from the GRU results.

The values are the medication for the past visits.

#### Ob and od

Ob = memory\_bank\_contribution = softmax(query . memory\_bank\_transpose). Memory\_bank

Od = key-store contribution = memory\_bank . (key\_value\_store\_values). Softmax(key store keys . query)

#### Sigmoid Output

We apply a sigmoid out to each of the outputs to scale them between 0 and 1. Anything above 0.5 is assumed 1 and anything below 0.5 is assumed 0.

#### Combined Loss

We calculate the losses from the model after evaluating each visit in each patient . There are 2 different losses that we look at:

1) **Binary cross entropy loss**  
We calculate the loss between the target output and the expected medication vector both in a binary form with array index representing medication feature.

2) **Muti label margin loss** This is used to maximize probability . We construct a target loss consisting of -1 and the medical feature value . Example : [3,4,5,-1,-1,-1]. We use this to calculate loss on sigmoid(model\_ouput)

The loss is combined to a single loss: loss = 0.9 \* Loss\_BCE + 0.01 \* Loss\_MML

#### DDI Score

The features of medications sigmoid output will be fed into the ddi\_score function to calculate drug-drug interaction scores. We will aim to minimize this function along with the loss of the predicted medications.

# Results

# A graph with blue lines Description automatically generated

A graph with a line

Description automatically generated

A graph with a line

Description automatically generated

A graph with blue lines

Description automatically generated

A graph with a line

Description automatically generated

# Discussions -Change

The GAMENet paper was reproducible, and we have shown the work above, however, the loss function that we have employed confirms the trainability of the Model, and we are converging. The paper has used a combined loss function, which penalizes the model for increasing DDI. This will be included in the next phase.

The code that came with the paper was well modularized, and defined, and made it easy to reproduce.

The Model definition, and the design was also a bit confusing. The NeuralNetwork that became the GAMENET augmented Seq2Seq model is implemented in one network, which makes the Network a bit complex.

In the next phase, we will:

* Update our loss function.
* Modularize the Neural Network, to facilitate easier implementation.
* Perform some model comparisons.

GameNet (Epoch 8)

| Metric | Result |
| --- | --- |
| DDI Rate | 0.057 |
| Jaccard | 0.44 |
| PRAUC | 0.675 |
| AVG\_PRC | 0.630 |
| AVG\_RECALL | 0.595 |
| AVG\_F1 | 0.592 |

# Model Comparisons

### Logistic regression

#### Creating dataset :

For Linear Regression , the feature dictionary created for GAMENET was reused and the features were taken from a “medical code : numerical value” mapping to a muti hot encoding form .

*Example:*

*Input :  
Total drugs : 5  
DRUG\_3 : 3,  
DRUG\_4 : 4  
DRUG\_5 : 5  
drugs\_vector\_for\_visit\_X = [2 ,3 ,4]*

*Output :  
multi\_hot\_encoding\_input : [0 , 0 , 1, 1, 1]*

#### *Results*

| Metric | Result |
| --- | --- |
| DDI Rate | 0.0782 |
| Jaccard | 0.4087 |
| PRAUC | 0.6739 |
| AVG\_PRC | 0.6730 |
| AVG\_RECALL | 0.5224 |
| AVG\_F1 | 0.5669 |

## Leap:

LEAP stands for "Learning to prescribe" and is a method designed to make effective and safe treatment recommendations, particularly in the context of multimorbidity, where patients suffer from multiple diseases simultaneously, necessitating complex medication strategies.

Leap models work similar to NLP seq2seq models. They are trained on EHR data, and attempt to learn the recommendations based on previous patient visits. They include attention mechanisms, to include the patient history in the recommendations.

The Leap model seems to have a lower ddi score, though the precision and Recall don’t do as well as the GameNet model.

*Results*

We trained the Leap model on the EHR data (without the ddi data), and got the following results

| Metric | Result |
| --- | --- |
| DDI Rate | 0.053 |
| Jaccard | 0.3551 |
| PRAUC | 0.5142 |
| AVG\_PRC | 0.4930 |
| AVG\_RECALL | 0.5489 |
| AVG\_F1 | 0.5006 |

## Discussion model comparison

# References

[1] Johnson, A., Pollard, T., & Mark, R. (2016, September 4). *Mimic-III clinical database*. MIMIC-III Clinical Database v1.4. <https://physionet.org/content/mimiciii/1.4/>

[2] Offsides and twosides (NSIDES v0.1) – tlab. (n.d.). http://tatonettilab.org/offsides/

[3] Shang, J., Xiao, C., Ma, T., Li, H., & Sun, J. (2019). GAMENet: Graph Augmented MEmory networks for recommending medication combination. *Proceedings of the . . . AAAI Conference on Artificial Intelligence*, *33*(01), 1126–1133. https://doi.org/10.1609/aaai.v33i01.33011126

[4] Zhang, Y., Chen, R., Tang, J., Stewart, W. F., & Sun, J. (2017). Leap. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/3097983.3098109