Predicting 30-days Unplanned Readmission Using Deep Learning Model with Electronic Health Records

ShanShan Jiang, Yin Yuan, Zheng Fu
Georgia Institute of Technology, Atlanta, Georgia

Abstract

Accurate prediction of unplanned hospital readmission could effectively reduce the readmission risk and the cost of healthcare system. In this project, we proposed a novel method to present the clinical notes of patients' electronic health records (EHRs) using open-source NLP research library built on PyTorch. And we demonstrated that deep learning algorithms using this representation were able to accurately predict 30-days unplanned readmission without disease-specific information. Our final deep learning model could reach the average accuracy about 0.85 on the unseen validation set. A video representation of this project could be found at the following link: https://youtu.be/5CoJ2MEU-58.

Introduction

30-days unplanned readmission is a common and costly problem for U.S. hospitals. It was defined as the admission to a hospital within 30-days after discharge. Medicare Payment Advisory Committee has reported that 17.6% of hospital-admitted patients were readmitted within 30 days of discharge. It gave rise to $17.9 billion Medicare spending per year and 76% of them could be avoidable[1]. Thus the ability to predict 30-days readmission would be important to develop interventions to reduce this risk as well as the cost of healthcare, especially for the patients characterized by chronicity and high rates of recurrence of a certain disease. The challenge here is how to predict individual 30-days readmission risk accurately due to the heterogeneity of the patient population along with the complicated health conditions of patients.

In recent years, several pioneering studies were published focusing on 30-days hospital readmission prediction on the basis of patient Electronic Health Records (EHRs). Efrat Shadmi et al. developed a Preadmission Readmission Detection Model (PREADM) using a preprocessing variable selection step with decision trees and neural network models. The selected features then were feed into a multivariable logistic regression model to predict the all-causes 30-days emergency readmissions. They reported c-statistic was 0.70 in the derivation cohort (two-thirds) and of 0.69 in the validation cohort (one-third)[2]. Oanh Kieu Nguyen et al. conducted an observational cohort study using EHRs data from six hospitals. They first estimated the univariate relationships between readmission and each of the candidate predictors, and the significant predictors (p-value < 0.05 based on the univariate test) were entered in a multivariate logistic regression model using stepwise backward selection. They also performed several sensitivity analyses to confirm the robustness of their model. The final predictive model could interpret additive effects of in-hospital complications, clinical trajectory, and stability on discharge on the risk of 30-days readmission with c-statistic of 0.69 (95% CI: 0.68-0.70)[3]. Charles A. Bailli et al. tested 30 models on predicting 30-days readmission with retrospective and prospective validations, and the best model has c-statistics = 0.640, sensitivity = 0.53, specificity = 0.74 and F score = 0.350[4]. Courtney Hebert et al. employed multivariable analysis with stepwise removal and created three risk disease-specific risk prediction models and a combined model. The models performed well on a random sample validation cohort with AUC range from 0.73 to 0.76[5]. Sharath Cholleti et al. developed a knowledge driven pre-processing step followed by random forest model to compute risk for 30-days readmission of patients in ten disease categories with AUC score no less than 0.70[6]. Xiongcai Cai et al. built a Bayesian Network model for predictions of length of stay, mortality, and readmission using EHRs. This non–disease-specific model could archive accuracy of 80% and area under the receiving operating characteristic curve (AUROC) of 0.82 in terms of predicting readmission[7].

All the studies mentioned above were based on structured EHRs data. Using unstructured EHRs free-text data to predict 30-days readmission is still a challenge work due to the confounding interactions between disease progression and readmission, along with the incompleteness, noisiness and heterogeneity of EHRs free-text data. Rumshisky A. et al. extracted inpatient psychiatric discharge narrative notes from a cohort of individuals admitted to a psychiatric inpatient unit with a principal diagnosis of major depressive disorder. Then they trained a 75-topic Natural Language Processing (NLP) model that identified groups of words associated with topics discussed in a document collection. The AUC scores of the baseline model, baseline+1000 words model and baseline+75 topics model are 0.618, 0.682 and 0.784, respectively[8]. Alvin Rajkomar et. al. proposed a predictive deep learning model with EHRs data based on the Fast Healthcare Interoperability Resources (FHIR) format. It has high accuracy on in-hospital mortality (area under the receiver operator curve [AUROC] across sites 0.93–0.94), prolonged length of stay (AUROC 0.85–0.86), all of a patient's final discharge diagnoses (frequency-weighted AUROC 0.90), yet low accuracy on 30-days unplanned readmission (AUROC 0.75–0.76)[9]. As far as we know, currently there is no established model to combine both NLP models and deep learning algorithms on predicting 30-days unplanned readmission with good performance in terms of accuracy. So the primary goal of this project, is to extract meaningful signals from unstructured text and refine them with deep learning models to accurately predict the 30-days unplanned readmission. The beauties of deep learning models are their end-to-end learning capabilities, and we wish the information derived from our models could intervene on behalf of patients at risk of unplanned readmission.

Methodology

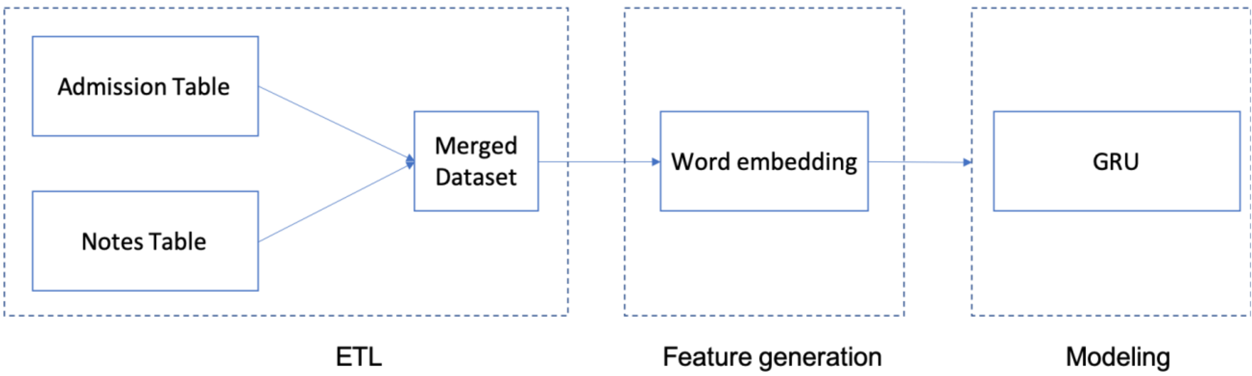


Figure 1. System framework of our project

There are three major components of our workflow, the ETL step, Feature generation step and Modeling step. The entire framework is shown in Figure 1.

1. Data pre-processing and ETL

MIMIC-III is a public-access database contains EHRs data from over 40,000 patients who were admitted to Beth Israel Deaconess Medical Center from 2001 to 2012[10]. We downloaded the entire MIMIC-III dataset and focused on the following two tables:

1.1 ADMISSIONS

This table contains admission and discharge dates. In order to protect the confidential information of patients, the dates in this table were shifted and will be internally consistent for the same patient, but randomly distributed in the future. There are 58976 recorded admissions in total in this table. The distribution of different admission type is illustrated in Figure 2, and we only consider unplanned admissions.

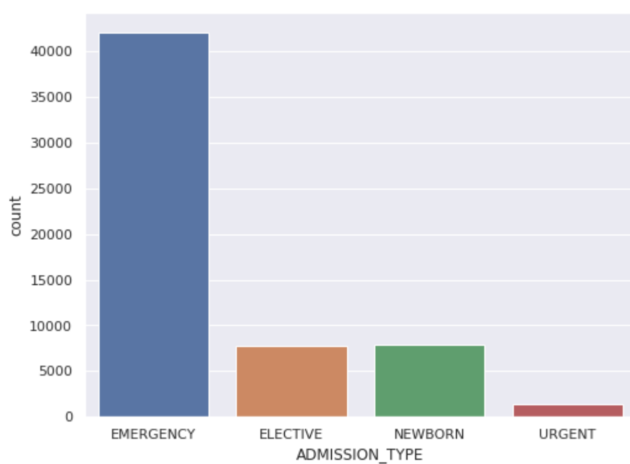


Figure 2. Distribution of different admission types.

1.2 NOTEEVENTS

This table contains the clinical notes of each hospitalization. The clinical notes are in free-text format.

The strings in “ADMISSIONS” table were converted to date format and since our goal is to predict unplanned readmissions, hereby we nullified the “ELECTIVE” next admissions. After that we generated the labels from “ADMISSIONS” table. Such table is self-joined by “subject id” with shifting each patient's next admission to the current admission row. As a result, we can get the patients’ next admissions. We ignored the planned admissions and only consider unplanned admissions, i.e. emergency admissions. Based on the unplanned admission, we computed the days until next admission. Only readmission less than 30 days are conceded as positive samples (1) otherwise negative samples (0). For training data, we loaded the “NOTEEVENTS” table, and filter the discharge summaries. For clinical notes, we only used the “Discharge summary” note and removed “NEWBORN” subtype of “Discharge summary” note. Such note will be converted by NLP methods to generate data for machine learning (see next section). Finally, we combined the data and labels, split the dataset into training/testing/validation sets. Since we have more negative samples than positive samples in the dataset, we oversampled the positive samples to balance the data. The distribution of the samples is shown in Figure 3.

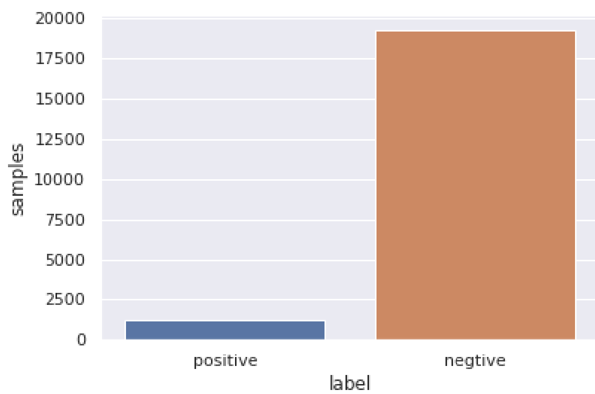


Figure 3. sample distribution of positive and negative labels

2. Converting text data to numeric vectors

Since the medical notes in MIMIC-III are so long, the deep learning models could not use them directly and we have to convert the free-text data into numeric vectors. Here we just extract some relevant information, which is related to the description of the discharge data like Discharge Medications, Discharge Disposition, Discharge Diagnosis, Discharge Condition, Discharge Status and Discharge Instructions. After extraction, the first step is to do tokenizer to break description sentences into tokens. Second, removing words that do not contain practical meaning using nltk library. Considering the memory usage and limitation, we only kept the first 60 tokens after removing the stop word. Then we use the method called ElmoEmbedder() from allennlp library to extract features, which is the most popular feature engineering method recently. The ELMo subcommand allows user to make bulk ELMo predictions. Given a pre-processed input text file, this command outputs the internal layers used to compute ELMo representations to a single (potentially large) file. The input file should be previously tokenized, whitespace-separated text, one sentence per line[11]. The idea of ELMo is to dynamically adjust the word embedding on the basis of the context. Here we employed a pre-trained ELMo language model with two layers of LSTM. The first layer of LSTM is in charge of forward embedding and the second layer of LSTM is for backward embedding. The context-free encoding of tokens were performed by a CNN model, and the outputs of three layers (one CNN + bi-LSTM) were scaled to 1024 dimensions. Thus each token finally will be presented by a 3 * 1024 tensor and feed to the deep learning model for predicting 30-days unplanned readmission.

3. Deep learning model for predicting 30-days unplanned readmission.

Figure 4 shows the architecture of the deep learning model we used for predicting 30-days unplanned readmission. Here we employed a RNN with GRU structure, and the ELMo features of each token within one sentence was feed to each step of the RNN. There is only one hidden layer with 8 nodes inside the GRU. We tested different number of

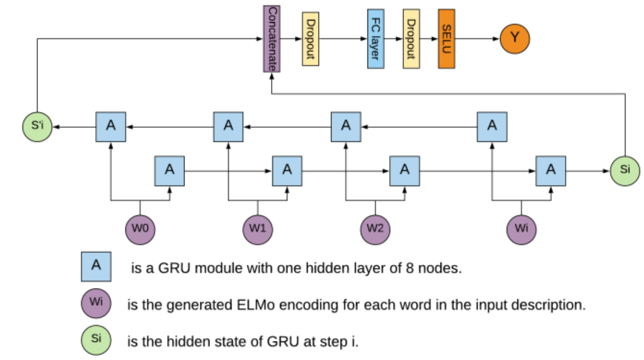


Figure 4. The architecture of deep learning prediction model.

hidden layers and one layer with 8 nodes has the best generalization. We could see from Figure 4 that the GRU has bidirectional topology. While outputting the outcomes, we concatenated the hidden states of last steps on both directions, and then feed it to a dropout layer to reduce overfitting followed by a fully connected layer. The outputs of the fully connected layer were passed to a dropout layer and feed to a SELU activation function to generate the binary decision. We tuned up the model with 30000 training observations, 15334 validation observations and 15334 testing observations, and the following are the optimized parameters of our final model:

ELMO_DIM = 1024, HIDDEN_DIM = 8, OUTPUT_DIM (FC layer) = 2, N_LAYERS = 2, BIDIRECTIONAL = True, DROPOUT = 0.5

4. Evaluating Model Performance

(1) Accuracy: (True Positive + True Negative) / (True Positive + True Negative + False Positive + False Negative)

(2) AUC score: area under receiver operating characteristic (ROC) curve.

Results

In order to investigate the optimal topology of the deep learning model on predicting 30-days unplanned readmission, we compared different deep learning model topologies as following. In each comparison, we used our final model as reference to compare models generated by adjusting only one parameter of the proposed configuration.

1. Comparison bi-directional with single direction of GRU RNN network (Figure 5 and 6).

2. Comparison on different number of internal layers used in GRU (Figure 7, 8 and 9).

3. Comparison on different number of nodes used in the internal layer of GRU (Figure 10 and 11).

4. Comparison on different dropout value used in the FC layers of our network (Figure 12 and 13 and 14).

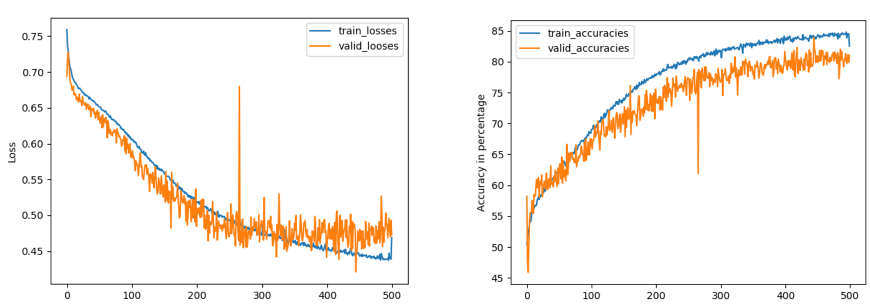


Figure 5. Performance of single directional GRU RNN model.

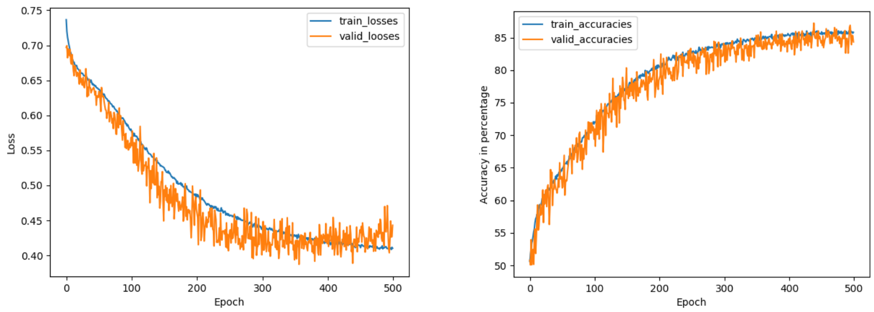


Figure 6. Performance of bi-directional GRU RNN model.

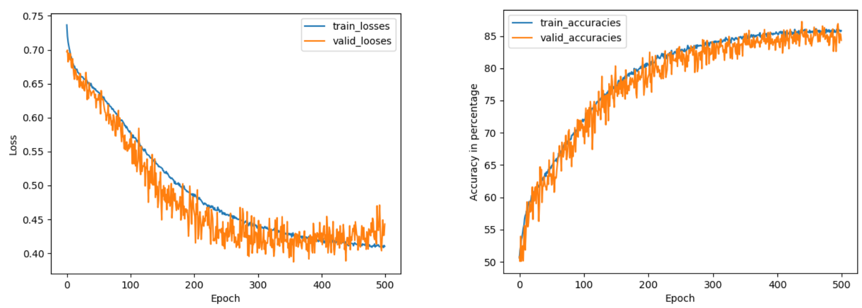


Figure 7. Performance of GRU RNN model with one internal layer.

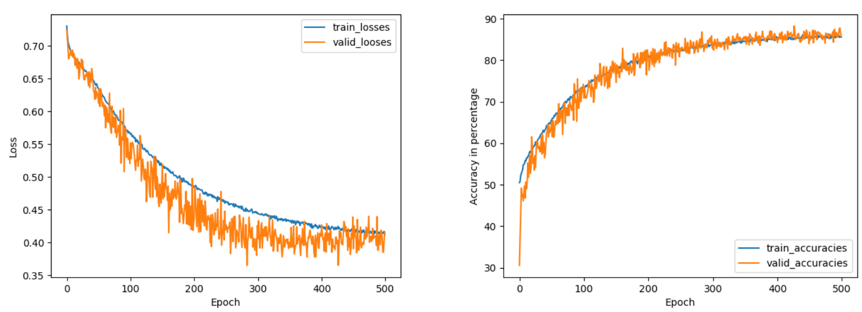


Figure 8. Performance of GRU RNN model with two internal layers.

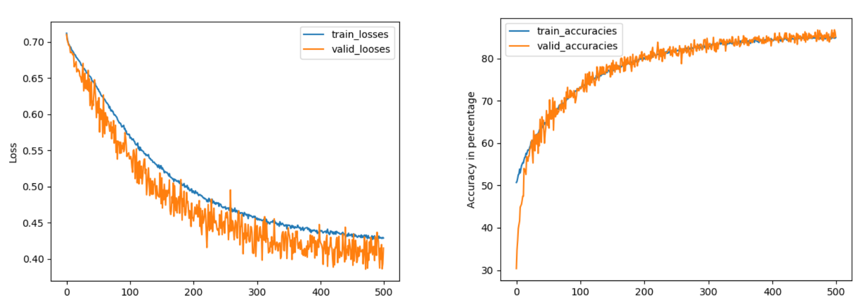


Figure 9. Performance of GRU RNN model with four internal layers.

Discussions

Figure 5 and 6 revealed the bi-directional GRU RNN model achieved better performance. Comparing with single directional topology, bi-directional model might have better capability to learn internal relationship and context

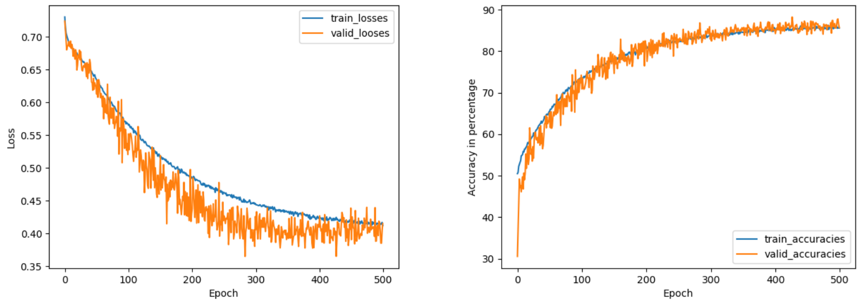


Figure 10. Performance of GRU RNN model with 8 nodes in internal layers.

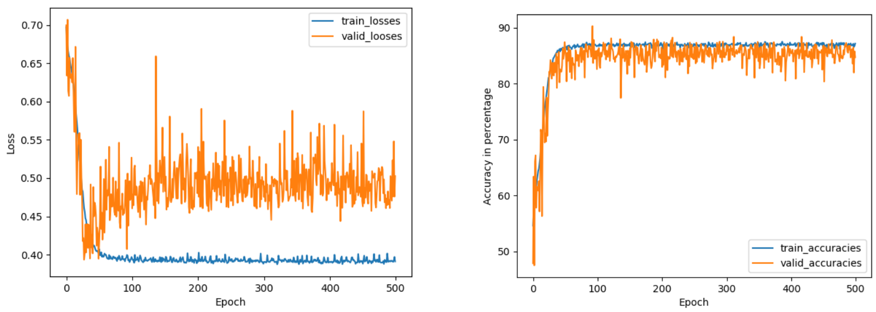


Figure 11. Performance of GRU RNN model with 128 nodes in internal layers.

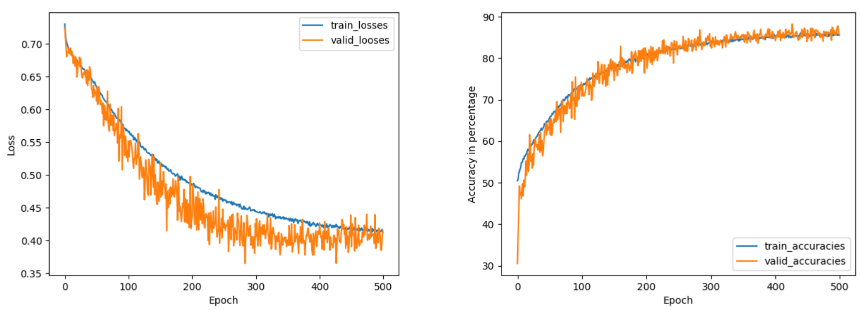


Figure 12. Performance of GRU RNN model with dropout rate = 0.5.

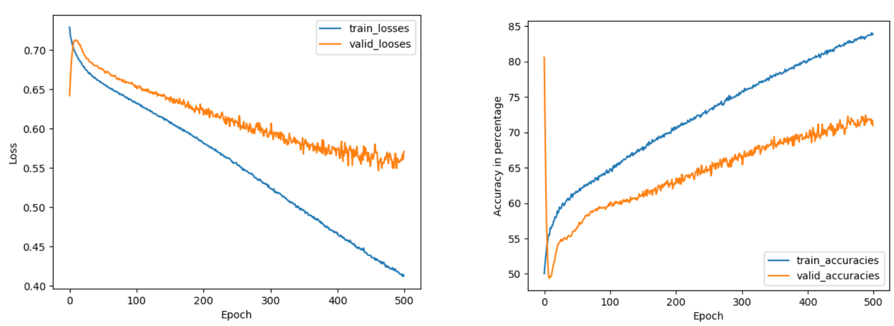
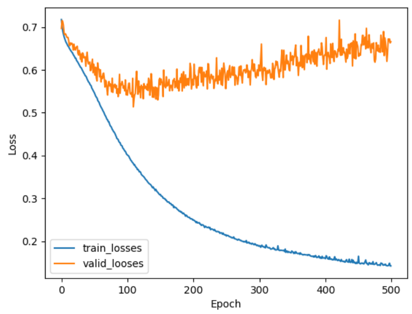


Figure 13. Performance of GRU RNN model with dropout rate = 0.8.

information among different words in the medical descriptions. Figure 7, 8 and 9 indicated that the GRU RNN model with 2 internal layers has the best performance in terms of both loss and accuracy. Using only one layer makes the



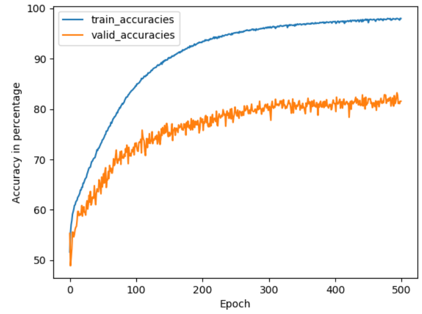


Figure 14. Performance of GRU RNN model with dropout rate = 0.2.

network is a bit simple which didn't dig enough information from input data. By using 4 layers, network didn't converge as fast as the one of using 2 layers. Figure 10 and 11 show that using 128 nodes will cause moderate overfitting if we checked the loss curve, and using 8 nodes could fix the overfitting issue dramatically. Figure 12, 13 and 14 reveal that dropout = 0.5 is appropriate value to use in our case. Dropout = 0.8 will simplify a network too much. Thus, the network is not able to learn the complex medical diagnosis description. Using dropout = 0.2 will make network overfitting the training data.

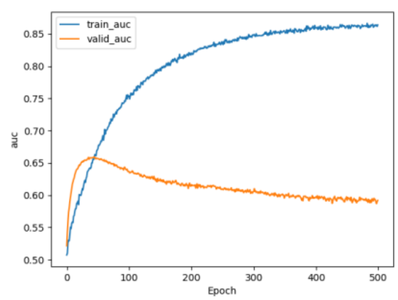


Figure 15. Performance of GRU RNN final model in terms of AUC score.

Figure 15 shows the AUC score changing with the Epoch numbers and we could see the model reaches it maximum AUC score at 0.65 around 50 epochs. One possible reason for this phenomenon is the dataset are highly unbalanced. Although we have already over-sampled the positive observations, we still do not have sufficient information to train the model to distinguish the true positive and false positive. Thus although the model has good accuracy, it may perform well on specificity yet lack of sensitivity. In the future we might develop different data balancing strategies to improve the prediction accuracy on positive observations.

Conclusion

In this project we proposed a framework to extract, transform and load patients re-admission information from MIMIC-III database, followed by start-of-arts ELMo algorithm to embed the word and feed the outputs to a well-tuned-up GRU RNN model. We trained such model with 30000 observations and tested on 15334 unseen observations to obtain average accuracy score more than 0.85 after 400 epochs. Our GRU RNN model has demonstrated that combining NLP techniques and machine learning algorithms could effectively predict the 30-days unplanned hospital readmission using unstructured free-text data. Instead of just developing and employing single deep learning model, there are lots of rooms to improve the prediction accuracy using cutting-edge NLP embedding methods with ensemble different deep learning models, which will be the primary object of our future work.

References

1. lvennan CK, Eapen ZJ and Allen LA. "Hospital readmissions reduction program". Circulation. 2015;131:1796-1803.

2. Shadmi E, Flaks-Manov N, Hoshen M, Goldman O, Bitterman H, Balicer RD. “Predicting 30-days readmissions with preadmission electronic health record data”. Med Care. 2015 Mar;53(3):283-9.
3. Nguyen OK, Makam AN, Clark C, Zhang S, Xie B, Velasco F, Amarasingham R, Halm EA. “Predicting all-cause readmissions using electronic health record data from the entire hospitalization: Model development and comparison”. J Hosp Med. 2016 Jul;11(7):473-80.
4. Baillie CA, VanZandbergen C, Tait G, Hanish A, Leas B, French B, Hanson CW, Behta M, Umscheid CA. “The readmission risk flag: using the electronic health record to automatically identify patients at risk for 30-days readmission”. J Hosp Med. 2013 Dec;8(12):689-95.
5. Hebert C, Shivade C, Foraker R, Wasserman J, Roth C, Mekhjian H, Lemeshow S, Embi P. “Diagnosis-specific readmission risk prediction using electronic health data: a retrospective cohort study”. BMC Med Inform Decis Mak. 2014 Aug 4;14:65.
6. Cholleti S, Post A, Gao J, Lin X, Bornstein W, Cantrell D, Saltz J. “Leveraging derived data elements in data analytic models for understanding and predicting hospital readmissions”. AMIA Annu Symp Proc. 2012;2012:103-11.
7. Cai X, Perez-Concha O, Coiera E, Martin-Sanchez F, Day R, Roffe D, Gallego B. “Real-time prediction of mortality, readmission, and length of stay using electronic health record data”. J Am Med Inform Assoc. 2016 May;23(3):553-61.
8. Rumshisky A, Ghassemi M, Naumann T, Szolovits P, Castro VM, McCoy TH, Perlis RH. “Predicting early psychiatric readmission with natural language processing of narrative discharge summaries”. Transl Psychiatry. 2016 Oct 18;6(10):e921.
9. Rajkomar A, Oren E, Chen K, Dai AM, Hajaj N, Hardt M, Liu PJ, Liu X, Marcus J, Sun M, Sundberg P, Yee H, Zhang K, Zhang Y, Flores G, Duggan GE, Irvine J, Le Q, Litsch K, Mossin A, Tansuwan J, Wang, Wexler J, Wilson J, Ludwig D, Volchenboum SL, Chou K, Pearson M, Madabushi S, Shah NH, Butte AJ, Howell MD, Cui C, Corrado GS, Dean J. "Scalable and accurate deep learning with electronic health records". NPJ Digit Med. 2018 May 8;1:18.
10. Johnson AE, Pollard TJ, Shen L, Lehman LW, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, Mark RG. "MIMIC-III, a freely accessible critical care database". Sci Data. 2016 May 24;3:160035.
11. Matthew P, Mark N, Mohit I, Matt G, Christopher C, Kenton L, and Luke Z. “Deep contextualized word representations”. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, Volume 1 (Long Papers), pages 2227–2237, New Orleans, LA.

Team member contribution

1. Shanshan Jiang

1. Getting medical notes extracted from upstream spark data generation step. Using NLP approaches to process the notes, including cleaning the notes, tokenizing the notes, removing stop words from notes.
2. To start with, building and training a baseline model using random forest algorithm. Generating bag of words features out of pre-processed notes from last step.
3. For building the deep learning algorithm:
3.1. Implementing an approach to extract ELMo features from our extracted medical notes using a pretrained ELMoEmbedder which is from a 3rd party python package.
3.2. Given ELMo features extracted, preparing training, validation and testing data for the proposed deep learning algorithm and building dataloaders for these data which will be used in the deep learning algorithm.
3.3. Building a GRU based neural network and getting it trained using generated ELMo embeddings.
3.4. Optimizing the network by comparing results of many different hyperparameter configurations. Finalizing the model for submission by choosing the one with best performance.

2. Yin Yuan

1. Playing one of the key roles in methodology evaluation.
2. Creating data pipeline to prepare training data, using spark;
3. Visualizing data and summarizing data statistics to present the distribution of key variable.

3. Zheng Fu

1. QC and validating codes.
2. Writing 40% of the contents of the proposal.
3. Drafting the whole paper's structure, writing the abstract, introduction, 90% of the methodology, 50% of the results, 50% of the discussions, conclusions and references of both draft and final report.
4. Generating all slides for presenting this project.
5. Presenting this project and uploading the video to YouTube.