NLP to Identify High Risk ICU Readmission Patients

Sam Falk SJF374@NYU.EDU
Center for Urban Science + Progress
370 Jay Street, 13th Floor
Brooklyn, NY 11201

May 10, 2020

Abstract

Deep learning models have improved significantly over the years. The evolution of natural language processing(NLP) has shown a lot of options to tackle medical analysis bottle necks that have existed for decades. The invention of BERT (Bidirectional Encoder Representations from Transformers) has opened the doors for more wide spread adoption of NLP in higher risk areas that could effect patient care. Now that the architecture of training in both left and right contexts has proven it's worth, there are opportunities to think of how it might improve the workflow within the hospital, specifically the task of care coordinators to identify patients at risk for readmission. This study explores that possibility with the use of the MIMIC III dataset and BERT models. The study is attempting to evaluate if it is plausible for clinic notes to predict readmission of a patient into an ICU. In the end, the best model had an AUC of 0.64. This did not meet the benchmark of a manual evaluation from a care coordinator, but does show promise for further iteration on the model.

1. Purpose

Currently in healthcare there is a race to find better ways at creating easy-to-analyze structured data from detailed notes or learnings from the physician interaction with the patient. For example, the mCode initiative created by MITRE is re-imagining how the data itself could be collected at the bedside, specifically for cancer patients. One of their major goals is to create data that is ”Standardized and collected in a computable manner, so it can be aggregated with data from many other patients and analyzed for best practices” (ASCOPost, 2019). In the other direction, Cerner's newest product, Chart Assist, is using voice recognition to identify major chart areas that already exist and populate those based on the bedside human interaction (Siwicki, 2020).

These and many more projects show the breadth and depth of investment in natural language processing (NLP) of medical notes, but why is it so important to explore? While examining the role of a care coordinator, you'll find responsibilities to review charts and identify patients who are at risk for readmission or worsening conditions. This is with the intent to provide medical interventions. The process is very manual and much of their chart review is based on pre-established inclusion criteria found in the free text of doctors notes. Despite the human-driven intervention, care coordination has been seen as an effective way to lower readmissions. “Analysis suggests that [care coordinator] intervention improved the likelihood of not being readmitted by some 22 percent” (Bronstein et al., 2015).

2. Introduction

This analysis explores visit note data that has already been collected to better understand the correlation between the content of those notes and the probability of readmission to an ICU. This can be used to help care coordinators better identify patients at risk, save hospital resources and in turn increase quality of care with previously mentioned interventions. Prior studies have linked readmissions to a positive correlation with length of stay, cost of care, and mortality (Brown et al., 2014). Specifically, my experience working with hospital leadership and addressing patient volumes through data science over the past three years has influenced my hypothesis. The hypothesis is that using deep learning methods on clinical notes can be used identify a patient at risk for readmission within 90 days of the last ICU visit.

3. Related Work

There has been much research dedicated to studying readmissions, both in a theoretical sense and also a technical capacity. In respect to Deep Learning Natural Language processing, Google has proven to be a source of innovation in the field. In the top GLUE benchmarks leader board, Google teams can be seen twice in the list and ranks forth overall (Warstadt et al., 2018).

In 2018, a study was released by Google evaluating the power of using deep learning on electronic health records (EHR). The conclusions from this research are used as benchmarks for the analysis below. Additionally, they included benchmarks from two real world hospitals. This is important to consider as a benchmark because of the project goal to improve the arduous process of care coordinators efforts to identify patients as risk for readmission. (Rajkomar et al., 2018).

Also the model this paper will focus on is BERT (Bidirectional Encoder Representations from Transformers), which once again was developed by Google and many new iterations of the model is included in the leader board as well. (Devlin et al., 2018). This model has been seen as superior to many of the other NLP deep learning models because of the architecture that trains text both forwards and backwards. Instead of only predicting the next wors in a series of strings, it instead predicts the word before and the words after. The bidirectional nature leads to a higher predictability power, compared to another model like ELMo (Peters et al., 2018). ELMo was originally considered for this study, but after further research many sources indicted BERT is a more effective model for this type of work. “Compared to ELMo, BERT is deeper in how it handles contextual information due to a deep bidirectional transformer for encoding sentences”(Si et al., 2019).

4. Data

To evaluate the hypothesis, MIMIC-III (Medical Information Mart for Intensive Care III) data set(Johnson et al., 2016) was used as a source of patients and their EHR records. This data set is a ”large, 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.” It is important that this data is large to ensure the model has enough cases to learn from.

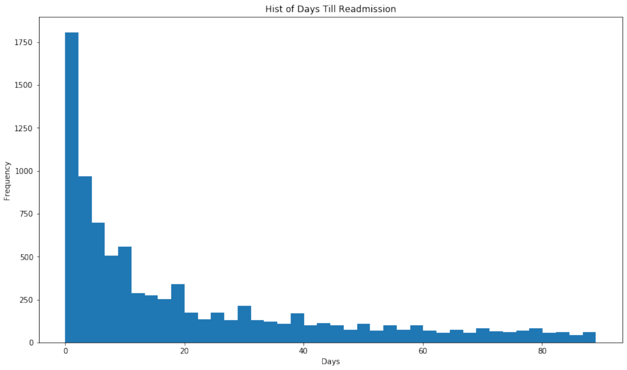

Figure 1: Histogram of the time elapsed to next admission. Can be seen that it is skewed right towards smaller amount of days between admissions to ICU

This data set contains 24 csv files, but noteevents, icustays, and patients are the tables of most interest. Data munging and cleaning consisted of steps to obtain the label of re-admittance, combine records to ensure each record was comprehensive of the all the notes for the visit, and filter to relevant records. The model is meant to predict if a single admission will result in a readmission after, so each record being trained is based off a single ICU stay. To determine if the stay resulted in a readmission, it was checked if the same patient had another record after the existing ICU stay record within 90 days.

The choice of 90 days was an intentional one, some other research has used 30 or 60 day readmission cut offs, but this would limit the the positive cases. Increasing to 90 days allowed for greater power of the model. In Figure 1 you can see the distribution of the days between stays. Most applicable records fall in the 20 days or less category.

This left about 14 percent of visits with a positive value. There are 8,711 records of visits that resulted in the patient returning to the ICU after their visit out of a total of 61,134 visits within the ICU data set. This imbalance is addressed later in the paper when considering the results of the model.

In some cases, there were multiple note records per ICU visit, so all of these notes were concatenated together prior to NLP. At times, after this occurred there were very long records, so truncating the records in those cases was important as well. The cut off for truncating was at the 90th percentile of text length: 164168.

	Train	Test	Val	Total
Readmission	5,177	1,795	1,739	8,711
No Readmission	31,504	10,432	10,487	52,423
Total	36,681	12,227	12,226	61,134

Table 1: Breakdown of records to labels and their stratification across the train, test, and validation sets. These are the counts passed into the unbalanced models.

The data set was split into training, test and validation sets. The records were divided randomly into 60 percent for the training set and 20 percent for each of the test and validation sets. Table 1 provides actual ICU stay quantities per each set.

5. Methods

To transform the data set into a usable form the text must be tokenized for use by BERT. In some ways this makes the process for this project considered semi-supervised learning. Though the final target is a clearly defined, the processing of the text is a bit more abstract. In other words, it will translate the text into digestible arrays encoded into an array of various numerics. To the human eye these numerics seemingly don't represent much, but do link back to the core words.

This project leveraged the pre-trained BERTtokenizer that is essentially a WordPiece tokenizer. (Wu et al., 2016). The WordPiece tokenizer does what it implies and handles words in pieces. It is especially useful to use this method for medical text as many of the words are not common, but when broken down can be understood. Take for example "fetal tachycardia", which is likely not a common word or phrase in the Google training set. When the word is broken down, boundary symbols are added before each sequence to assist in the encoding process to ids. The output of tokenizing is something similar to this:

'fetal', 'ta', '**chy', '**card', '**ia'

This method is an ideal balance between word-focused and character-focused models because it will decide how common the word is and then change the evaluation method based on that decision. There are multiple parameters passed into the tokenizer to create the final variable

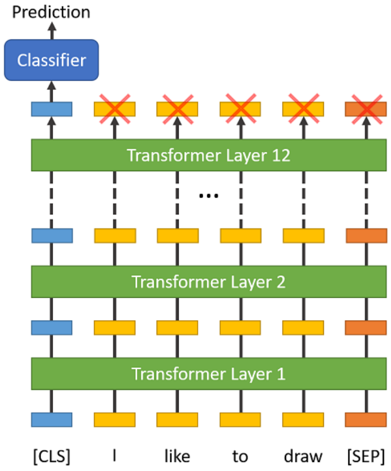

Figure 2: Visual representation of the 12 layer BERT small lowercased model

input to the model. Of most importance is the maximum length of the final output. This was set to 264 to handle memory issues on the remote machine, manage time constraints of the project and mitigate possible over fitting. Processing any tokenized text that is shorter than 264 ids is then padded to ensure the data set is in the same format across records.

Now that the input to the model is defined, the model itself can be trained on the data. The BERT model is a pre-trained 'bert-base-uncased' model. This means that the model is the smaller of the two possible pre-trained ('base' or 'large') and processes specifically lower-cased letters. Next, is to use supervised learning methodologies as a classifier to output the probability of readmission. This classifier will includes sigmoid based functionality, softmax to ensure that the outcomes are clearly defined.

The model architecture consists of 12 layers that were trained on lower cased English text. Using a model that already exists and is validated on large sets of data saves time, increase the effectiveness of the model, and allows for further fine tuning. This is called transfer learning. Figure 2 demonstrates in a visual component of what is occurring in this model as it trains. The illustration came from an article about fine tuning a BERT model with the change in hyper parameter. (McCormick and Ryan, 2019). Modifying hyper parameters is important to ensure the model is not over or under fitting in the training procedure. Learning rate for ADAM and the number of epochs are parameters I adjusted in different iterations of the model. ADAM learning rate determines how quickly the model will learn and it is a delicate balance of quick learning and over fitting. The values that were attempted in this project include: 5e-5, 3e-5, and 2e-5. Epochs also contribute to this balance and the models used 3, 5, 6 and 25 epochs. To use one epoch would mean that the entire dataset is fed into the model in both directions only one time. when scaling and using more epochs, the data is split up and fed into in smaller quantities.

Another hyper parameter that was not adjusted in the development was the batch size, which is the number of records that were processed before the model is updated. This variable was kept this at size 16, because of memory issues with any bigger size on the NYU remote work space. In the results, there is more detail around the most effective use of each of these hyper parameters.

6. Results

To evaluate the effectiveness of the model, accuracy on the test set and validation epoch loss was utilized to track how the model changes with each epoch. To consider the use case of hospitals, focusing on minimizing false negatives in predicting the test set is imperative. Ideally the care coordinators would want to intervene when a patient is at risk. To possibly miss a patient in the model can result in a patient safety or liability issue.

At first, the entire dataset mentioned in the DATA section was used to pass into the model. This meant that the labels were very unbalanced, with 14 percent of records resulting in a readmission. The resulting model had very high accuracy rates. One would expect this to be a positive, but upon further inspection many

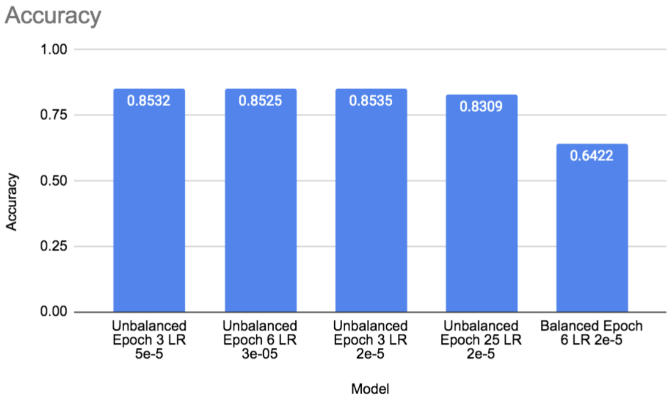

Figure 3: Test set accuracy scores of the various models run and their hyper parameters. The unbalanced models performed best in this measurement, but really they were no better than guessing because the negative class has a larger ratio to the positive class.

of these models had increasing loss or predicted very small quantities of the positive class. Simply picking the negative class at all times, results in a very high accuracy rate of 0.8532. Unfortunately that makes the model completely unusable in the real world scenario, because no patients would ever be flagged for care coordinators or receive medical intervention.

Of the models trained with this unbalanced data set, there were two models that seemed to have the ”best” performance. One with three epochs and a learning rate of 2e-5 resulted in an accuracy of 0.8535, a relatively stable loss of the validation set, and correctly identifying only 3 percent of the positive cases. The other is one with the same learning rate of 2e-5, but instead with 25 epochs. In this case, the loss increased by more than 0.02 across the training of the 25 epochs and ended with an accuracy of 0.8309. Though this accuracy on the test set is less than the previous, the model actually predicted 13 percent of the records with a positive class. Still though, if stopping here it would have been suggested that the null hypothesis could not be rejected. This is largely because the notion that guessing is about the same as the model holds true. A closer look at the AUC of .501 also confirms the null hypothesis cannot be rejected, The null hypothesis in this case being that clinical notes can not be used identify a patient at risk for readmission within 90 days of their last ICU visit.

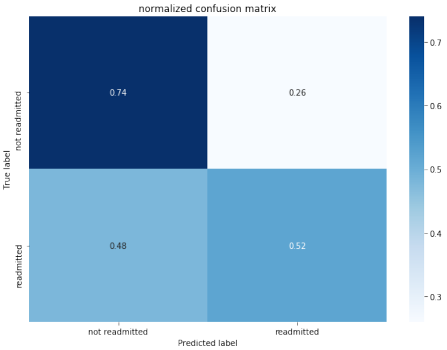

Figure 4: The model with balanced classes was able to identify about half of the positive cases, compared to the unbalanced model identifying 13 percent of the positive cases.

Following the training of these models, the dataset itself was revisited. This time, removing record for patients that expired between their their current stay and 90 days after their release. This is important because those patients would never have an opportunity to actually be readmitted. Also, The data set was balanced to have an equal amount of positive and negative cases. This left 6,769 records in both the positive case and negative case sets. With a balanced dataset, the hope is that the model will have a better opportunity to not over fit and instead truly learn the difference between positive and negative cases. The way the data was tokenized stayed the same. The total of 13,538 cases were split into training test and validation sets, similar to what was done before with 60 percent in training and 20 percent in each validation and test sets. The outcome of models with this balanced dataset, though much lower test set accuracy scores, showed some other interesting insights. The best of them performed with a learning rate of 2e-5 and 6 epochs. This time, the accuracy was 0.6422, but the AUC improved significantly with 0.6417. The model even was able to predict 52 percent of the positive class in the test set. Figure 4 shows the normalized confusion matrix for this model on the test set.

	Hospital A	Hospital B
At Discharge	0.77 (0.75–0.78)	0.76 (0.75–0.77)
Baseline (mHOSPITALc) at discharge	0.70 (0.68–0.72)	0.68 (0.67–0.69)
Unbalanced Epoch 25 LR 2e-5	0.501	
Balanced Epoch 6 LR 2e-5	0.642	

Table 2: Despite major improvements in the model when balancing the classes, the model still does not meet benchmarks for the hospital manual care coordination standard.

When comparing these outcomes to the bench marked hospital performance, it isn't as good as an actual care coordinator, nor the model created in that study (Rajkomar et al., 2018). On the other hand there is real promise to further effort to improve this model. Table 2 shows the comparison of the outcomes of this project compared to the 2018 study and figure 5 displays the models ROC curves themselves.

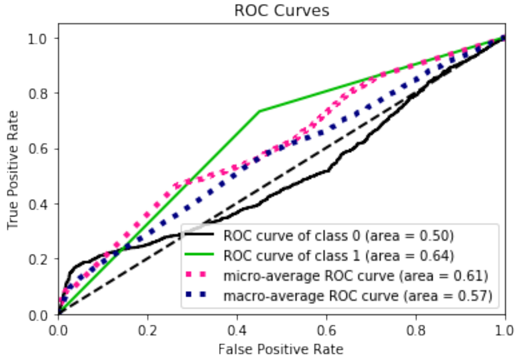

Figure 5: The model has improved overall but is still struggling to predict the positive class. This ROC curve graph is for the balanced model with 6 epochs and a learning rate of 2e-5.

7. Discussion

Diving deeper into handling the balanced classes did help the model, but there is much need for improvement. As I look forward to future work that could be done on this project, I see some possibilities into examining the weights of classes passed into models to help with the unbalanced data set. Also, considering an increase in allocated days for readmission definition could balance the target labels. Possibly increasing to 120 days may be of interest. Furthermore, examining the BERT tokenizer could improve the model. Teams out there have been working to specifically use transfer learning based off of medical data sets. This would ensure the common words as mentioned before are considered medically related.

It was proven in this project that tuning the hyperparameters was useful and there may be more opportunity there. Possibly increasing the batch size if more GPUs could be obtained. Many decisions in this development were based on the limitations of NYU's virtual machines, but to remove those limitations may open up more possibilities for model improvement.

The model as it stands does not seem sufficient to place in the hands of care coordinators, but it does seem plausible to create a product off a model that could be easy to use for EHR data analysis. I envision an outcome of a properly performing model in practice would assist in interventions of patients who are at risk.

Besides the extensive references to class and literature review, the sole active contributor to this project was Sam Falk.

8. Mechanics

Overall, jupyter notebooks were used in the NYU Prince environment. The project took shape by first cleaning the data and passing it through the tokenizer. Then, The output of the tokenizer plus the labels were used to train the BERT model. There were many iterations of munging the data and training a model with different hyper parameters, but across all instances model evaluations were performed to analyze train, test, validation accuracy scores, epoch loss, and AUC curve score.

References

The ASCOPost. 2019 asco: mcode, a core set of common cancer data standards, established - the asco post, Jun 2019. URL https://ascopost.com/News/60120.

Laura R. Bronstein, Paul Gould, Shawn A. Berkowitz, Gary D. James, and Kris Marks. Impact of a Social Work Care Coordination Intervention on Hospital Readmission: A Randomized Controlled Trial. *Social Work*, 60(3):248–255, 05 2015. ISSN 0037-8046. doi: 10.1093/sw/swv016. URL https://doi.org/10.1093/sw/swv016.

Erin G Brown, Debra Burgess, Chin-Shang Li, Robert J Canter, and Richard J Bold. Hospital readmissions: necessary evil or preventable target for quality improvement. *Annals of surgery*, 260(4):583–591, 10 2014. doi: 10.1097/SLA.0000000000000923. URL https://pubmed.ncbi.nlm.nih.gov/25203874.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

Chris McCormick and Nick Ryan. Bert fine-tuning tutorial with pytorch. 07 2019. URL http://www.mccormickml.com.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M. Dai, Nissan Hajaj, Michaela Hardt, Peter J. Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, Patrik Sundberg, Hector Yee, Kun Zhang, Yi Zhang, Gerardo Flores, Gavin E. Duggan, Jamie Irvine, Quoc Le, Kurt Litsch, Alexander Mossin, Justin Tansuwan, De Wang, James Wexler, Jimbo Wilson, Dana Ludwig, Samuel L. Volchenboum, Katherine Chou, Michael Pearson, Srinivasan Madabushi, Nigam H. Shah, Atul J. Butte, Michael D. Howell, Claire Cui, Greg S. Corrado, and Jeffrey Dean. Scalable and accurate deep learning with electronic health records. *npj Digital Medicine*, 1(1):18, 2018. doi: 10.1038/s41746-018-0029-1. URL https://doi.org/10.1038/s41746-018-0029-1.

Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association*, 26(11):1297–1304, 07 2019. ISSN 1527-974X. doi: 10.1093/jamia/ocz096. URL https://doi.org/10.1093/jamia/ocz096.

Bill Siwicki. At himss20, cerner will be talking ai-powered voice technology, Feb 2020. URL https://www.healthcareitnews.com/news/himss20-cerner-will-be-talking-ai-powered-voice-technology.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *arXiv preprint 1805.12471*, 2018.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144, 2016.