Team B4 Project Draft

Team Members: Devin Cortese, Ethan Tran
CSE-6250: Big Data Health, Fall 2023

Introduction

Patient phenotypes play a critical role in the healthcare industry in practice and in research. Therefore, it is critical to accurately assign patient phenotypes to advance the medical field and to make critical decisions for patients in practice. Our paper [1] defines phenotyping as: "Mining electronic health records for patients who satisfy a set of predefined criteria [1]". Manually mining electronic health records is a time consuming and often inaccurate task. Therefore, automating this process is extremely valuable to researchers and practitioners. Our paper [1] aims to push the boundaries of phenotype automation by using advanced machine learning techniques to effectively assign phenotypes to patients by using clinical notes from electronic health records. More specifically, the authors aim to convert these clinical notes into numerical representations by essentially performing dimensionality reduction or data compression on the notes using deep learning architectures. To assess their effectiveness, the paper [1] utilizes the resulting dense representations as classification inputs for downstream machine learning tasks, such as predicting patient comorbidities, including asthma, diabetes, and obesity. Our goal for this project is to reproduce the results achieved in this paper [1] to determine if this method of phenotype automation is practical and to potentially improve it.

Scope of Reproducibility

This paper [1] hypothesizes that we can map patient clinical notes to a lower dimensional numeric representation, using the patients' actual visit outcomes as a derivation standard for creating these new representations. Additionally, the authors believe that this methodology will produce representations with a quality that is sufficient for follow-up machine learning tasks. They hypothesize and confirm that this methodology achieves better performance when predicting patient comorbidities versus using uncompressed text and traditional dimensionality reduction techniques. We attempt to reproduce these experiments by recreating the model architecture used to extract the reduced patient representations and using them to predict comorbidities using a baseline support vector machine and traditional dimensionality reduction. We also aim to improve the representations by running experiments with different model architectures. For this project draft, however, we only focus on reproducing the patient representations using the model architecture provided by the paper [1]. Downstream tasks will be discussed in our final submission.

Methodology

Data Description

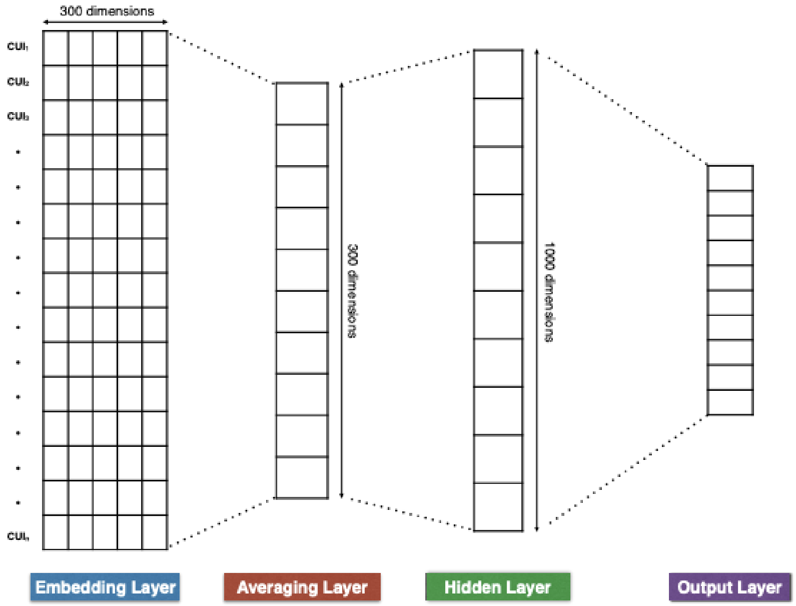
Similar to the paper [1], we are using a subset of tables from the MIMIC-III database [2] to extract our patient representations. Specifically, we use a table containing over 2 million clinical notes for different patients to extract numeric representations. One major difference with our implementation is that we are not mapping this text to UMLS concept unique identifiers (CUI's), as the paper [1] used a special software called cTAKES to accomplish. This software has a large learning curve, so we attempted to utilize an alternative tool called pyMetaMap to map text to the UMLS Metathesaurus, but the process was too inefficient and would have taken days to complete. Rather, we convert the text to tokens, such that each token is a lowercase word with no punctuation or numerical value. We then map our set of words to indices, so that each unique word has its own integer value as input to our model. Tokens that do not appear at least 100 times in the dataset are removed, along with stopwords. We also ensure that each patient's set of notes are concatenated before our tokenization, so that we maintain our patient-level representation with a single sequence of tokens for each unique patient in our dataset. After concatenation, we remove patients that have over 10,000 tokens, similar to the paper [1]. This leaves us with 37,789 unique patients, which is different from the paper [1] at 44,211, likely due to our differing tokenization methodology.

Each patient is aligned with a sequence of multi-hot encoded billing codes, representing our supervision labels. These billing codes include ICD9 diagnostic codes, ICD9 procedure codes, and CPT codes, pulled from tables in MIMIC-III [2]. Once again, we mirror the paper [1] and only keep billing codes that have at least 1000 appearances across all patients. This presents us with a large-dimensional multi-label classification problem with 174 unique billing codes for each patient. Finally, we randomly split our final patient-level dataset into a train (80%) and validation (20%) set. Note that we do not use a test set, as our post-training testing methodology is the downstream tasks to be presented later.

Model Description

We attempted to replicate the representation model architecture as close as possible to achieve a reliable baseline for additional experiments. Therefore, our first layer is an embedding layer, creating embeddings for each unique token each with a length of 300. For our architecture, our embedding layer only uses randomly initialized embeddings, whereas the paper [1] experimented with embeddings initialized from Word2Vec [3] as well. We average these embeddings together and pass them to a hidden layer with

1000 dimensions and a ReLU activation. The output from this layer will be our patient-level representations, and we will test them using our downstream tasks in the future. For these representations to learn, we need to pass them to another linear layer and a sigmoid activation function with 174 dimensions, matching the number of possible billing codes for any given patient. A visualization of the model architecture can be seen below, and note that this diagram is pulled directly from the paper [1], so the input CUI's would be replaced with our tokens.



To train this architecture on our data, we utilize a RMSprop optimizer and binary cross-entropy loss, since this is a multi-label classification problem. We train on 10 epochs for our initial build as opposed to 75 due to computational requirements, and utilize identical parameters as the paper [1]. This includes a batch size of 50 and a learning rate of 0.001.

Computational Implementation

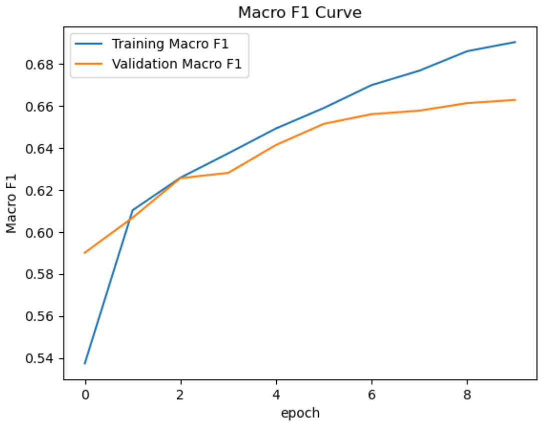
As mentioned previously, we have not used any GPU's to train our representation model yet. Given our solid performance with only 10 epochs already, this may not be necessary, and the effort to spin up extra computational resources may not be worth it. We will discuss the training performance in a later section.

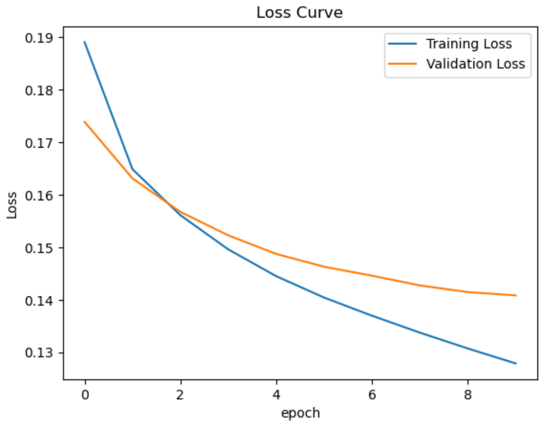
Code

Given that the goal of this project is to replicate the experiments performed in this paper [1], we heavily leaned on the provided repository's data preprocessing script. There were some alterations required from us given our differing tokenization and data collection methods, however. On the other hand, we did not utilize TensorFlow for model architecture and training like the paper's repository. Rather, we used Pytorch and the boilerplate code provided by previous homeworks to increase flexibility for future experiments. Our repository (https://github.com/dcortese6/CSE6250-Project) is also publicly available containing required setup instructions in the README documentation.

Results

Since this is a multi-label classification problem, we are not using accuracy to assess performance of our representation model on the validation set. Instead, we use Macro F1, which considers both class-based precision and recall and handles any class imbalances that might exist in our labels. Overall, we were able to achieve a score of 69% on the train set and 66% on the test set, with a minimum loss of 0.13 (train) and 0.14 (validation), which is a significant improvement from the paper [1]. Even with fewer patients in our dataset and using random initial embeddings, our tokenization method so far seems to be more effective than mapping patient notes to CUI's, but we will test the hidden layer embeddings with the comorbidity prediction models later to make a final determination.





Discussion

It is worth reiterating that, while our current model performance metrics point to an improvement over the original paper [1] implementation, we need to extract the output from our hidden layer and input them into our comorbidity prediction models to assess their value. This will serve as our main focus for the rest of development. We will also continue to improve the current representation model now that we have the baseline model from the paper [1]. One immediately noticeable improvement we can make is to increase the number of epochs, as our score and loss plots do not show asymptotic behavior, indicating room for improvement. We could also experiment with other dataset manipulations, like increasing or decreasing the number of required tokens for a patient to be in the dataset. These manipulations will likely be necessary due to our unique tokenization methodology.

References

[1] Dligach, Dmitriy, and Timothy Miller. “Learning patient representations from text.” *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 2018, https://doi.org/10.18653/v1/s18-2014.

[2] Johnson, Alistair E.W., et al. “Mimic-III, a freely accessible Critical Care Database.” *Scientific Data*, vol. 3, no. 1, 2016, https://doi.org/10.1038/sdata.2016.35.

[3] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781

[4] Khalafi, Sahar, et al. “A hybrid deep learning approach for phenotype prediction from Clinical Notes.” *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 4, 2023, pp. 4503–4513, https://doi.org/10.1007/s12652-023-04568-y.