Team B4 Final Project

Devin Cortese[1], Ethan Tran[2]
CSE-6250: Big Data Health, Fall 2023
Georgia Institute of Technology
dcortese6@gatech.edu[1], etran49@gatech.edu[2]
Code Repository
Presentation Video

***Abstract*—This paper discusses our re-implementation of the paper *Learning Patient Representations From Text* [1] for our final project of CSE-6250: Big Data Health from the Georgia Institute of Technology. We attempt to validate the experiments performed in this paper with some modifications and to determine whether the paper is ultimately reproducible or not.**

I. 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, so 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 of unseen clinical notes 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 upon it.

II. 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 attempted to improve the representations by running experiments with different model architectures. However, we only discuss reproducing the patient representations using the model architecture provided by the paper [1] due to computational requirements for complicated architectures. Given these representations, we can evaluate them by pushing a new set of clinical notes through our model to extract new representations and input them into a support vector machine (SVM) to predict patient comorbidities. We compare the accuracy of this "dense" SVM to similar evaluations on the uncompressed text and reduced text, along with the downstream task results produced in the paper [1].

III. Methodology

A. Dataset 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.

To evaluate representations, similar to the source paper [1], we use discharge summaries from the Informatics for Integrating Biology to the Bedside (i2b2) Obesity challenge dataset [5]. These summaries will act as our "unseen" data to input into our representation model. The output is the "compressed" version of this text, which maintains the same label mapping as the input text. Each summary is labeled as one of "present", "absent", or "questionable" for 16 different comorbidity categories. Our preprocessing for this data is similar to our MIMIC-III [2] preprocessing, such that we use the same extracted vocabulary and tokenization mapping with the same limitations as discussed previously. This results in approximately 1200 discharge document samples mapped with labels for each comorbidity.

B. 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 evaluate them using our downstream tasks. 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 in Fig. 1, and note that this diagram is pulled directly from the paper [1], so the input CUI's would be replaced with our tokens.

To evaluate the performance of this architecture, we use three different support vector machines. One will take as input the preprocessed i2b2 data [5] tokens in a "Bag-of-tokens" format. The second model will utilize a singular value decomposition form of the same tokens as a form of "compressed" data. Finally, the third model utilizes the output from the hidden layer of our representation architecture as another form of "compressed" input.

C. Computational Implementation

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. We were able to increase the number of epochs to 30 with our computational capabilities, but there was some overfitting as described later.

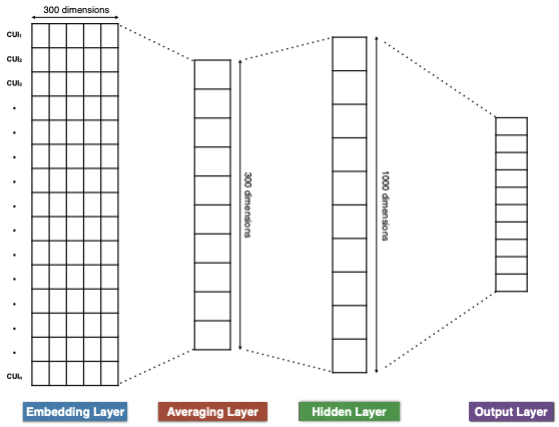

Fig. 1: Model architecture for baseline representation architecture, pulled directly from source paper [1].

Significant attempts were made to stand-up accelerated infrastructure to train more complicated architectures and to map CUI's to clinical notes. Even with Apple Silicon Chips, training was able to converge within our epoch range. However, with proper GPU capabilities, we are confident that CUI's could be efficiently mapped to notes and deeper networks could be built to extract representations. For what it's worth, our attempt at cTakes and MetaMap mapping was initially successful, but only 10 documents were being processed per hour, with around 45,000 documents to process in total, making our tokenization method much more favorable in the short term. Given our results, as discussed later, we begin to question whether this mapping may even be necessary in the long run.

D. 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 to maintain controlled experiments for our model. However, there were some alterations required from us given our differing tokenization and data collection methods. Additionally, 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 evaluation code also closely mirrors the original paper's repository. We closely follow this code to ensure that our evaluation is accurate along with our preprocessing. By controlling for our preprocessing and evaluation techniques, we can ensure that any difference in our results and the original source paper's amounts to the difference in the model training or tokenization methodology.

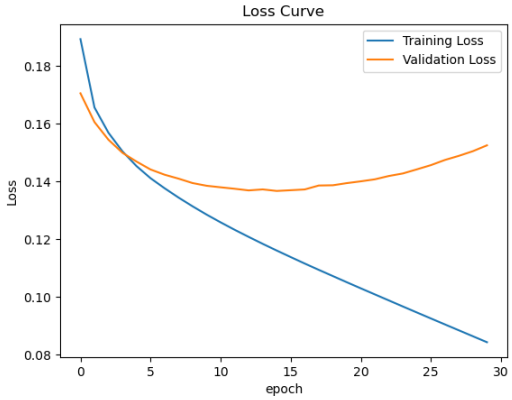

Fig. 2: Average Binary Cross-Entropy Loss for 30 epochs during on training and validation sets.

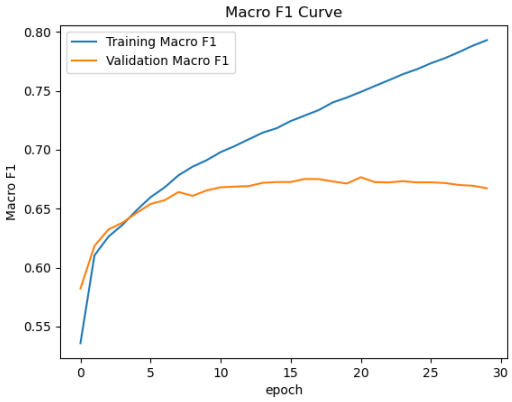

Fig. 3: Average Macro-F1 score for 30 epochs during on training and validation sets.

Our repository is also publicly available containing required setup instructions in the README documentation.

IV. Results

Since our representation model 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 80% on the train set and 66% on the test set, with a minimum loss of 0.09 (train) and 0.14 (validation). Clearly there is some over-fitting here as seen in Fig. 2 and Fig. 3, but the important evaluation is on our downstream tasks. Even with fewer patients in our dataset and using random initial embeddings, our tokenization method seems to be effective.

To assess the true effectiveness of our representation architecture, we compare the performance of our three baseline support vector machines by calculating their precision, recall, and F1 scores on the test i2b2 data [5]. We also compare these metrics with those of the source paper. The final results can be seen in Fig. 4. This table contains our scores (token method) to the left of the forward slashes and the paper's scores (CUI method) to the right (Token/CUI).

It is first worth noting that our Bag-of-Tokens and SVD Patient-Token models *slightly* under-perform when compared to the paper's CUI models. While some diseases, like diabetes, maintain the same performance or even out-perform the CUI methods, the overall averages tell a different story, with the tokenization models showing about a 5-10% decrease in performance across all metrics. Just from these results, we know that tokenization of clinical notes is overall not as effective as extracting CUI's when phenotyping patients, as there is clearly some information loss. This loss of information lowers expectation for our learned representation architecture. Assessing this version of our "compressed" token data on the learned representation support vector machine shows an over 10% decrease in performance when compared to the paper's CUI methods. While the paper has their learned representation baseline model out-performing the others, ours actually under-performs the other two models with the tokenization method. Therefore, we cannot confirm the original hypothesis from the source paper with our reimplementation.

V. Discussion

While our results clearly under-perform, we feel comfortable stating that the source paper is reproducible. To justify this conclusion, we point to the over-fitting of our representation model architecture. This proves that the architecture struggles to generalize the clinical notes when using the tokenization method, and our expectation from this was a less valuable patient representation extraction, as shown by our evaluation results. Additionally, the purpose of CUI's is to categorize concepts that are captured in these notes and to maintain the same information at a higher, more structured level. Our re-implementation at least shows that these CUI concepts are indeed effective, and that if we were to perform the proper mapping along with the use of pre-trained embeddings, we would be able to achieve similar results to the source paper. It would also be possible to improve upon it, with proper computational resources and new architectures not presented in the paper.

There were several difficulties when re-implementing this paper. The first obvious issue was CUI extraction. Attempts were made at this, but the process is extremely slow and unrealistic in the short-term, so the pivot to tokenization was made. Another difficulty was computational resources, but that is no fault of the source author. The source paper did a great job of explaining the architecture used and where they harvested the representations from in their model. This made it relatively easy to translate the architecture training from Tensorflow to Pytorch. A more detailed explanation of their pre-processing could have been made however, as there was some uncertainty during our data preparation as to how the notes were concatenated per patient. Additionally, their initial organization and folder structure of the datasets was a bit unclear. Ultimately, these hypotheses and experiments were very clearly communicated to the reader, which made this re-implementation exciting and valuable.

	Sparse Bag of Tokens/CUI's			SVD Patient-Token/ CUI Matrix			Learned Representation		
Disease	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Asthma	0.862 / 0.894	0.691 / 0.736	0.739 / 0.787	0.869 / 0.888	0.850 / 0.854	0.859 / 0.870	0.667 / 0.910	0.765 / 0.920	0.688 / 0.915
CAD	0.571 / 0.583	0.575 / 0.588	0.573 / 0.585	0.587 / 0.593	0.596 / 0.602	0.590 / 0.596	0.551 / 0.596	0.543 / 0.596	0.547 / 0.596
CHF	0.484 / 0.558	0.381 / 0.564	0.310 / 0.561	0.523 / 0.571	0.511 / 0.575	0.501 / 0.573	0.516 / 0.588	0.517 / 0.564	0.512 / 0.561
Depression	0.761 / 0.797	0.677 / 0.685	0.702 / 0.715	0.645 / 0.723	0.655 / 0.727	0.650 / 0.725	0.571 / 0.791	0.596 / 0.773	0.567 / 0.777
Diabetes	0.863 / 0.859	0.840 / 0.853	0.850 / 0.856	0.613 / 0.611	0.621 / 0.624	0.617 / 0.617	0.817 / 0.907	0.820 / 0.919	0.818 / 0.913
GERD	0.494 / 0.530	0.425 / 0.466	0.439 / 0.485	0.472 / 0.533	0.431 / 0.482	0.443 / 0.499	0.416 / 0.528	0.422 / 0.539	0.419 / 0.533
Gallstones	0.811 / 0.814	0.559 / 0.640	0.566 / 0.678	0.601 / 0.747	0.599 / 0.793	0.600 / 0.732	0.554 / 0.645	0.582 / 0.663	0.550 / 0.653
Gout	0.927 / 0.975	0.711 / 0.811	0.772 / 0.871	0.874 / 0.955	0.808 / 0.834	0.837 / 0.882	0.684 / 0.928	0.690 / 0.910	0.687 / 0.919
Hypercholesterolemia	0.739 / 0.781	0.738 / 0.784	0.738 / 0.782	0.724 / 0.789	0.727 / 0.793	0.724 / 0.790	0.696 / 0.865	0.698 / 0.868	0.696 / 0.866
Hypertension	0.726 / 0.680	0.647 / 0.650	0.670 / 0.662	0.660 / 0.711	0.690 / 0.763	0.671 / 0.728	0.631 / 0.825	0.658 / 0.879	0.640 / 0.847
Hypertriglyceridemia	0.980 / 0.933	0.620 / 0.679	0.683 / 0.748	0.545 / 0.580	0.565 / 0.610	0.552 / 0.591	0.547 / 0.604	0.662 / 0.650	0.541 / 0.621
OA	0.509 / 0.514	0.425 / 0.448	0.442 / 0.466	0.456 / 0.479	0.409 / 0.442	0.420 / 0.454	0.429 / 0.511	0.400 / 0.508	0.407 / 0.510
OSA	0.580 / 0.596	0.471 / 0.511	0.503 / 0.542	0.570 / 0.626	0.494 / 0.568	0.521 / 0.592	0.505 / 0.611	0.506 / 0.618	0.506 / 0.615
Obesity	0.759 / 0.825	0.732 / 0.791	0.736 / 0.798	0.781 / 0.883	0.749 / 0.844	0.754 / 0.853	0.731 / 0.872	0.732 / 0.873	0.731 / 0.872
PVD	0.563 / 0.594	0.470 / 0.542	0.500 / 0.564	0.500 / 0.599	0.474 / 0.557	0.485 / 0.576	0.440 / 0.568	0.462 / 0.599	0.450 / 0.582
Venous Insufficiency	0.863 / 0.797	0.599 / 0.797	0.618 / 0.675	0.673 / 0.669	0.758 / 0.757	0.704 / 0.700	0.544 / 0.638	0.605 / 0.717	0.581 / 0.665
Averages	**0.718 / 0.733**	**0.598 / 0.650**	**0.615 / 0.675**	**0.631 / 0.685**	**0.621 / 0.672**	**0.620 / 0.674**	**0.581 / 0.709**	**0.604 / 0.725**	**0.584 / 0.715**

Fig. 4: Precision, Recall, and F1 scores for each SVM baseline, with each input type, for each comorbidity. Our score (token method) is to the left of the forward slash and the paper's score (CUI method) is to the right (Token/CUI).

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.

[5] Uzuner, O. "Recognizing obesity and comorbidities in sparse data." Journal of the American Medical Informatics Association, vol. 16, no. 4, 2009, pp. 561–570, https://doi.org/10.1197/jamia.m3115.