Representation Learning of Electronic Healthcare Records for Downstream Prediction Tasks: Comparing Deep Techniques

Jennifer Koenig
University of Illinois Urbana-Champagne
jkoenig4@illinois.edu

ABSTRACT

In this paper, I compare the strengths and weakness of three deep techniques for representation learning of electronic health care records (EHR). I do so with the publicly available dataset MIMIC-III [3] and recently proposed standard benchmarks [2]. The use of existing tools allows us to focus on knowledge-gain in answering the question of how to best represent EHR data for predictive healthcare tasks. This makes the practical usability of existing tools a key component in their value to the medical community.

Keywords

Representation learning, electronic health care records, benchmarks, reproducibility, usability

1.INTRODUCTION

Due to wide-spread adoption of electronic health records (EHR), we currently have an abundance of digitalized patient data available for use. Distilling this information into meaningful representations is essential for gaining knowledge. Good representations can be used for various important health-care tasks such as mortality prediction, disease classification, and patient phenotyping.

Representation learning improves upon traditional vector-space techniques such as one- and multi-hot vector encoding or bag-of-words, by allowing us to represent large amounts of related heterogenous data in small, dense embeddings. Unlike traditional encodings, these vector embeddings retain the "meaning" of the data and its internal relationships Using this technique, we improve the performance and interpretability of models used for various tasks in healthcare.

Until recently, there were no widely accepted standard benchmarks for comparing different representations of EHR. However, a study made available in 2017, "Multitask learning and benchmarking with clinical time series data" [2] (referred to in this paper as "the Benchmark study"), proposes just that, by providing four downstream benchmark tasks trained on publicly-available electronic health data [3]. My study uses these public benchmarks as a baseline for exploring two additional, recently proposed deep learning methods for representing longitudinal EHR, namely Med2Vec [1] and ConvAE [4].

The exploration, development and comparison of improved representations of EHR is important, even critical, in the endeavor to provide reliable, quality healthcare. With this study, I attempt to contribute to this endeavor.

2.APPROACH/METRICS

In this section I will describe 2.1) the infrastructure I created, 2.2) the predictive comparison task, 2.3) the metrics used to evaluate each model's performance, 2.4) the data, and finally the models themselves (2.5).

2.1.Infrastructure

I ran all of the code on AWS EC2 instances with GPU and large storage capacity. The images I used for Benchmarks and Med2Vec came pre-installed with the necessary Python components, but for ConvAE I had to use a deep learning base image and install the requirements myself. Where necessary, I uploaded the MIMIC-III data files required by the algorithms directly onto the instance where the code was hosted.

For the Benchmark model, I used an AWS EC2 p2.xlarge instance pre-installed with TensorFlow 1.7, Python 3.6, CUDA 9.1 and NVidia drivers. It had one GPU core with and a Linux/Unix Amazon Linux 2018.03.0 | 64-bit (x86) operating system.

For Med2Vec (data processing), I used an AWS EC2 p2.xlarge instance pre-installed with Theano 1, Python 2.7, NVidia drivers and one GPU core on an Amazon Linux 2017.09.1 | 64-bit (x86) operating system. For training and evaluating the model, I used an AWS EC2 p2.xlarge instance pre-installed with TensorFlow 1.5, Python 3.6, CUDA-only NVidia drivers and one GPU core on an Ubuntu operating system.

For ConvAE, I used an AWS EC2 g4dn.xlarge instance: "Deep Learning Base AMI (Amazon Linux 2) Version 36.0", built with NVIDIA CUDA, cuDNN, NCCL, and GPU Drivers.

2.2.Predictive Task

The Benchmark study proposes four predictive benchmark tasks to be used as standards for comparing the performance of various models. Due to time constraints, I used only the phenotyping task. The learned patient representations of both Med2Vec and ConvAE can be applied to this task and thus compared.

The task consists of predicting the presence or absence of 25 crucial care-conditions[1] in the last sequence of each patient, given medical codes or concepts of all previous sequences. These sequences can either be discrete visits, as in the case of Med2Vec, or hourly events, as in the case of Benchmark and ConvAE.

The prediction is a multi-class classification task: a combination of 25 separate binary classification tasks. Performance on this task varies by care-condition, as some conditions are more difficult to predict than others.

[1] https://www.nature.com/articles/s41597-019-0103-9/tables/2

2.3. Performance Metrics

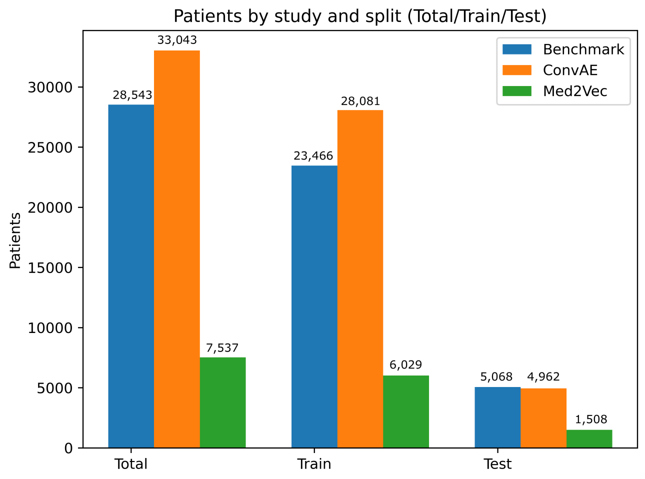
Average AUC-ROC is used to measure each model's performance on the prediction task per care-condition. ROC is a probability curve, and AOC is the area under that curve. Together, they tell us to what degree a model is able to distinguish between different classes.[7] AUC-ROC plots the false positive rate of a class prediction against its true positive rate: the closer that AUC to 1 is, the better the model is at predicting the correct class for a data sample. In this experiment, an AUC-ROC score 0f 0.5 indicates that the model is incapable of distinguishing the presence or absence of the care-condition in question.

Table 1 lists the 25 conditions, their prevalence in the data, and their average AUC-ROC performance scores on the Benchmark algorithm.

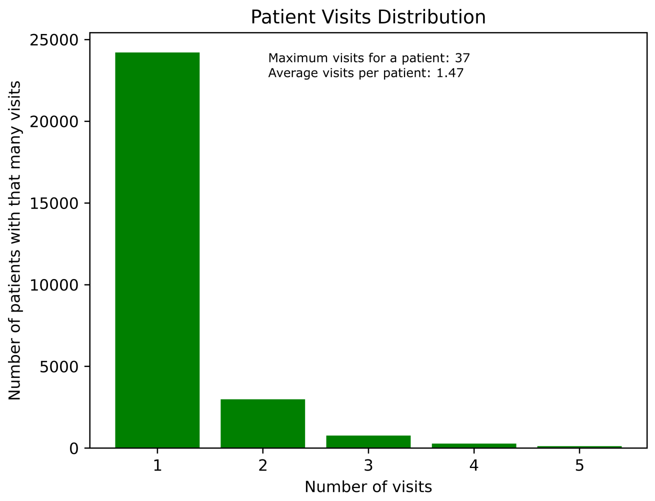
2.4. Data

2.4.1 Overview

The patient datasets for Benchmark and ConvAE were very similar, in that both studies could use data for patients that only had one visit. Med2Vec, however, had a much smaller subset, since it required a minimum of two visits for a patient if the data was to be used.



The absolute majority of number of visits per patient is one, as can be seen below. For this reason, ignoring data for patients with less than two visits reduced the available dataset by an extreme amount for Med2Vec.



Each study used chronological data in some way: Benchmark and ConvAE both used events ordered by timestamp (date and time) to represent patient sequences, while Med2Vec used visits ordered by date only. Because of this, Benchmark and ConvAE have an additional temporal level that is missing in Med2Vec.

Each study used ICD9 codes and vital signs in their creation of patient sequences. Benchmark integrated the values of vital signs (for example, "heart rate, 129.0), whereas ConvAE did not include values, just the names of the vitals. In Med2Vec, it was not clear from the study if the measurement values were included, although the study indicated that they integrated vital signs into their representations in some way.

Each study integrated categorical demographic into their data: ConvaAE included ONLY categorical demographic data, such as ethnicity and gender, whereas Benchmark and Med2Vec also included quantitive demographic data (specifically, age). Finally, each study used additional medical concepts in their patient sequences, which will be described in detail per study below.

Each study had a relatively equivalent prevalence of care-conditions in their populations, which was also reflected in the train and test splits. ConvAE's data, however, had unusually high incidences of non-hypertensive congestive heart failure and coronary atherosclerosis/other heart diseases, as well as somewhat more incidences of other liver diseases than in the other studies. (See Table 1).

2.4.2 Benchmark Data

The Benchmark study processes MIMIC-III data task-specifically. For the phenotype task, a total of 28,543 patients were processed and split into train and test sets.

The patient list consists of only those subjects that remained after event filtering, which excluded events that could not be matched to an ICU stay. Episodes of hospital admissions were aggregated into a time series for each patient, consisting of those filtered events belonging to a set of 17 preselected physiologic variables.[2] These variables come from chart, lab and output events from the MIMIC-III dataset.

The final processed data for training the phenotype model consist of patient admissions, each having a time-series file of events for 48 hours of the ICU stay associated with that admission, a length of stay column in hours, and 25 care-condition columns (with a value of 1 if the condition was diagnosed during that stay and 0 if not). The events, as stated above, consist of the 17 preselected physiologic variables and their values. For example, "heart rate, 129.0".

Since diagnoses are associated directly with a hospital admission and not an ICU stay, admissions were excluded that contained multiple ICU stays. This was done in order to reduce ambiguity: there is only one ICU-stay per hospital admission that a diagnosis can be associated with. [2]

The care-conditions (grouped diagnoses) chosen for classification are common to adults in the ICU and include acute, chronic and mixed conditions. Each condition is composed of multiple ICD-9 billing and diagnostic codes grouped into unambiguous disease categories. This grouping comes from from the Health Cost and Utilization (HCUP) Clinical Classification Software (CCS). Using the grouping instead of actual ICD-9 codes reduces noise, redundancy, and data size. An example of a condition used in the Benchmark study is "Diabetes mellitus with complications", which groups 57 related 3-digit ICD-9 codes.

2.4.2 Med2Vec Data

Med2Vec had a much smaller subset of MIMIC-III patient data to work with than the Benchmark study. This is because Med2Vec architecture represents temporality in the form of discrete visits (admissions). The timestamps of certain types of medical codes (such as chart events) are not represented in this model. In other words, patients have to have more than one visit to be included in the training, whereas Benchmark constructs a time series with hourly events (and so can include patients with just one visit).

The overwhelming majority of patients in the MIMIC-III dataset have just one visit. Due to this, the Med2Vec MIMIC-III subset had only 7,537 individual patients in total for nearly 20,000 visits. After splitting on a 80/20 ratio for training and testing, the training set only had slightly more patients than the testing-subsets of the Benchmark and ConvAE studies.

My use of Med2Vec included only ICD9 diagnoses as medical codes. I tried to re-create the benchmark study with both the complete ICD9 codes as well as only the 3-digit grouped ICD9 codes. The latter did not map one-to-one to the 25 care-conditions specified by the Benchmark study, but I took this into account in my code. Using the complete ICD9 codes resulted in data that could not be used in the AUC-ROC metric, as there were too many samples that had only one class (i.e. no true positives). Neither group was able to produce truly reliable results (more detail on this in section 3, "Experimental Results").

The original study claims to use vital signs but does not clarify how values from vital signs were represented. The original study also used both categorial and quantitive demographic data, which I did not integrate into my experiment for reasons I will explain in the "Experimental Results" section.

2.4.3 ConvAE Data

ConvAE could use the same set of patients as for the Benchmark model, due to the fact that temporality of medical concepts was not restricted (as it was in Med2Vec) to dates of admissions. In other words, patients with just one visit could also be included in the modelling. This resulted in a data size very similar to the size of the Benchmark patient dataset.

MIMIC-III diagnoses are organized so that they correspond to an admission date, and don't have any additional temporal concept. If a patient has only one visit, then the temporality component for that patient is missing, causing the data to be discarded by Med2Vec. In contrast, ConvAE is designed to take advantage of medical concepts that have a date AND a time. Concepts with a time component have a meaningful temporal order even within the context of a single visit. Such concepts include lab events and procedures.

These concepts, chronologically ordered and aggregated, compose the patient sequences that ConvAE uses in its model. Combined, the concepts create a vocabulary that is used during modelling for embedding the patient sequences.

I trained two separate models with different vocabularies. In the first model, I used (complete) ICD9 diagnoses, lab events and microbiology events, which resulted in almost 50 million events and a vocabulary of 15,746 unique concepts. In the second model, I only used ICD9 diagnoses, which resulted in a little less than 500,000 events and a vocabulary of 14,567 unique concepts.

For the first model, I reasoned that lab and microbiology events, when having been required for a patient, convey important information even when not associated to their measurement value. This is in contrast to vital signs, which are taken regularly for all patients: without a measurement value or indication of normality or abnormality, they don't convey much meaning.

The original study used vital signs. It was not clear from either the study or the publicly available code how the values for vital signs were to be integrated into the model. I reached out to the authors of they study and they informed me that vital sign measurement values are NOT included in the patient sequences. This confirmed for me my decision not to use them as medical concepts.

It was not clear either how demographic information was to be integrated. I was informed by the authors (rather late) that only categorical demographic information, such as ethnicity and gender, were included in the patient sequences. Due to the lateness of this information, I did not integrate it into the data I used for my experiment.

The reason I tried two different models was because the results of the first model, with diagnoses, lab, and microbiology events, were so poor that they were unusable. The results of the second model, with only diagnoses in its vocabulary, were more interesting. I will describe this in more detail in Section 3, "Experimental Results".)

2.5.Models

2.5.1 Benchmark Model

The Benchmark study offers seven baseline models for phenotype prediction. I chose to use only one, the standard LSTM, based on the fact that it is the simplest of the non-linear models (which included channel-based, multi-task, and deep supervision versions of LSTM along with various combinations of these).

LSTM is a type of RNN that captures long-term dependencies of sequential data. The time-series part of the preprocessed data is resampled into regular intervals. Redundant variable measurements are removed (the last measurement is used) and missing values are imputed with a pre-specified "normal" value for that measurement[3]. Binary masks are created for each input to discern real values from imputed ones. Categorical and binary variables are one-hot encoded and numeric inputs are standardized.

The result is 17 pairs of time-series for each ICU stay. Each consists of a binary value per variable (1 if the variable was observed, 0 if not) and the numeric value for that variable if observed. The concatenated pairs of time-series compose the input for the LSTM model. The output, or target variable, for the phenotyping task is a 25-element binary vector: one element for each care-condition described in section 2.4.2. The Benchmark standard LSTM model uses a single LSTM layer to take an instance of this input and output and process a hidden layer. This hidden layer and the learned weights can then used in a sigmoid equation to classify the presence or absence of each of the 25 care-conditions for a patient's last visit.

I used the publicly available code[4] provided with the Benchmark study to process MIMIC-III data, learn, and evaluate models for phenotyping. The full model uses chart, lab and output events in addition to (as in the sampling model) patients, admissions, ICD9 diagnoses, and ICU stays. I supplemented the original code with my own:

- code to map the care-condition "task number" to its medical codes and labels
- code to parse the results for display

[3] https://www.nature.com/articles/s41597-019-0103-9/tables/1

[4] https://github.com/YerevaNN/mimic3-benchmarks

- code to calculate various statistics on the post-processed data

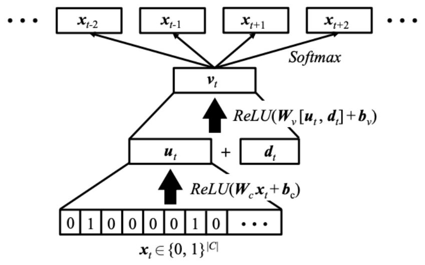
The strength of this model lies in the variety of information represented in the features and in the human-readability of the features before training. As described, each sample consists of length of stay and a multi-hot encoded condition vector. Patient information consists of temporal components including date and time, as well as values for vital sign measurements. The simplicity of running the models (once the infrastructure was set up), due to excellent documentation and the built-in evaluation metrics, make the Benchmark study easily reproducible. In fact, I was able to use the processed data from the Benchmark study as the basis for the ConvAE study.

The weakness of this model is the extra hardware (computation and storage) it requires to get it to run from beginning to end. Additionally, although the features are human readable, they are sparse and do not maintain the relationships between codes or visits, as dense-vector models do. Therefore, you cannot use the features generated by the Benchmark study to examine the similarity of diagnoses or of patients based on visit. However, due to the general nature of the features, they are usable in various prediction tasks besides phenotyping, including prediction of in-hospital patient mortality, patient decompensation, length of stay, and possibly other tasks of interest.

2.5.2 Med2Vec Model

Med2Vec is an algorithm proposed in the study "Multi-layer Representation Learning for Medical Concepts", authored by Choi et. al [1]. It learns distributed representations of visits and medical codes from EHR data. This representation has two levels: a sequential visit level, and a level of medical code co-occurrence within a visit.

First, one-hot encoded medical codes are embedded into a latent lower-dimensional space using ReLU activation. These are concatenated with patient demographic information to create an intermediate visit representation. Skip-gram is used on this intermediate representation to create a visit embedding consisting of medical code co-occurrences. The two objective functions for learning visit and code representations are combined in order to learn both from the same source of patient visit records.



The results are highly interpretable dense vectors that can be used for downstream tasks such as medical code prediction within a window context or patient phenotyping. These representations can be called distributed, since they consist of the same data features across multiple scalable and interdependent layers (i.e., temporal visits as well as the co-occurring codes within each visit).[5]

For Med2Vec, I used the publicly available code[6] to process MIMIC-III admissions and diagnoses into patient sequences. I then used these preprocessed sequences to feed into a publicly available TensorFlow implementation[7] of the Med2Vec architecture, supplemented and corrected with my own code.

Some of the alterations and additions I made to the model code include:

- code to split the data into training and test sets
- functions to prepare the model results for and evaluate them on AUC-ROC evaluation on 25 care-conditions
- functions for calculating various statistics such as number of patients and care-condition prevalence in test and train subsets, distribution of the number of visits and number of codes
- code for calculating Pearson's correlation coefficient and p-value on prevalence of care-condition and AUC-ROC score
- code for mapping Med2Vec-specific indices to MIMIC-III medical codes and care-conditions
- code for allowing the user to specify data files and model path on the command line instead of having to alter the hard-coding in the program files
- code for the objective function for the embedding cost. The original repository code threw errors when called, which probably explains why they did not use it:

$$\min_{\boldsymbol{W}'_c} \frac{1}{T}\sum_{t=1}^{T}\sum_{i:c_i\in V_t}\sum_{j:c_j\in V_t, j\neq i}\log p(c_j|c_i),$$

$$\text{where} \quad p(c_j|c_i) = \frac{\exp\left(\boldsymbol{W}'_c[:,j]^\top \boldsymbol{W}'_c[:,i]\right)}{\sum_{k=1}^{|\mathcal{C}|}\exp\left(\boldsymbol{W}'_c[:,k]^\top \boldsymbol{W}'_c[:,i]\right)}.$$

The original embedding cost equation maximizes the log likelihood of a code occurring in a visit given a code in the previous visit, for every combination of codes in the two visits (where one visit precedes the other) for a patient. Here, W'_c represents the non-negative code-weight matrix, where there is a column for every medical code. This cost needs to be added to the visit cost in order to learn visit and code representations simultaneously. The original TensorFlow implementation of this was incorrect, in that it added the visit cost to itself, instead.

Training the model had very high combined (visits and embedding) objective function costs which decreased only minimally over 20 epochs. The reduction remained minimal at even 200 epochs. This is likely due to the small set of patients available that had the required two or more visits. This minimal reduction in objective function occurred with both the model using 3-digit ICD9 codes (942 labels) and the one using complete ICD9 codes (4984 labels).

This model's strengths lie in the interpretability of its medical code representations, which maintain the "meaning" of the codes

[5] https://deepai.org/machine-learning-glossary-and-terms/distributed-representation

[6] https://github.com/mp2893/med2vec

[7] https://github.com/sdwww/Med2Vec_tensorflow/blob/master/Med2VecRunner.py

and their relationship to each other. This plays out well when clustering the code representations or projecting them onto two-dimensional space for understanding disease subtypes. The study authors show that individual coordinates of the code embedding matrix were able to convey significant insights about certain disease cohorts.

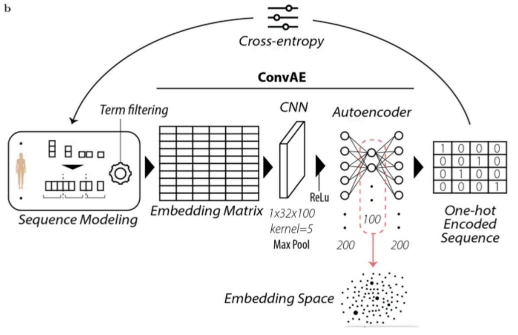
The weaknesses of the model lie in the visit representations when trained on too few data, which is the case with MIMIC-III under the filtering conditions described. The ability of the model to predict visit codes based on a previous window of visits is poor, which I will further detail in Section 3. With more data, the poor performance of this model on predicting visit codes would probably be improved.

An additional weakness is the lack of a fine-grained temporal concept for representing the sequences of medical codes that co-occur over a patient's history. In Med2Vec, this is limited to a visit level: date only. Because of this, much information is lost about how codes (for example, lab events) follow one another as the patient's condition evolves on an hourly basis. Also, the lack of measurement values for codes representing vital signs is a significant loss of information when representing patients, in addition to the fact that these kinds of information are not represented in a more fine-grained time sequence.

2.5.3 ConvAE Model

ConvAE is an unsupervised framework presented in the paper "Deep representation learning of electronic health records to unlock patient stratification at scale" [4]. It uses word embeddings, CNN and AutoEncoders (AE) on EHR to create patient representations.

b



The stated downstream purpose of these representations is patient stratification: determining latent patterns within patient groups, or identifying fine-grained disease subtypes at scale. However, the authors claim that "the representations aim to be domain free (i.e., not related to any specific task since learned over a large multi-domain dataset)" [4], and I was able to use them in the condition-prediction task described in section 2.2.

The paper shows that these low-dimensional latent vectors have a high success-rate in disentangling the heterogeneity of complex disorders, which makes them ideal, I believe, for the specific downstream task of predicting care-conditions. "While these representations were used to leverage disease subtype discovery, they can also be fine-tuned and applied to specific supervised tasks, such as disease phenotyping and prediction." [4]

ConvAE preprocesses patient data so that each patient is represented by a fixed-length sequence of aggregated, chronologically-ordered medical concepts. The chronology runs in order of date AND time. The medical concepts or the original study include ICD9 diagnoses, ICD10 diagnoses mapped back to their ICD9 versions, medications normalized to RxNorm, CPT-4 procedures, lab events normalized to their LOINC codes, (names of) vital signs, (categorical) demographic information, and medical text entities gleaned from unstructured clinical notes.

Each concept in the vocabulary is scored with TF-IDF, where the entire EHR is considered the document, document frequency is the number of patients that have at least one instance of the concept, and term frequency is the total number of times the concept occurs within that patient sequence. This is explained as the probability that a word occurs in the document, multiplied by the sum of normalized probabilities that it occurs in each patient sequence.

$$\mathbb{P}(w \in D) \sum_{p \in P} \mathbb{P}(w \in s_p) = \frac{\#\{s \in D;\ w \in s\}}{|D|}$$
$$\sum_{p \in P} \frac{\#\{w_i \in s_p;\ w_i = w\}}{|s_p|},$$

Concepts whose score falls outside of a certain threshold (10e-6) are dropped in order to remove uninformative concepts that are too rare to be informative. Duplicate concepts for a patient within an overlapping series of days are dropped, as are patients with less than 3 concepts. Finally the concepts within a time period are shuffled since they do not contribute a temporal property. The final concept vectors are padded to be fixed length.

The final patient sequences are first projected into a chronologically ordered embedding matrix. Because of this, they retain semantic (i.e. meaningful) relationships between medical codes as well as temporal information. CNNs are then used to model this temporal information by extracting local temporal patterns. AEs estimate this incoming subsequence by deriving an embedded patient representation. Finally, Softmax is used in a multi-class classification that reconstructs each initial one-hot encoded subsequence of medical terms from their encoded representations.[4]

It was not clear from the study or the code how the pre-trained word embeddings part of the model were created (i.e., on what text). I reached out to the authors in mid-April and got the information shortly before the deadline, that instead of using indices or codes for patient sequences, they used labels describing these codes. For instance, if a patient sequence contained a lab event with the the LOINC code 10381-2, they used the string "Target Cells-Blood-Hematology" as the index. After the patient sequences were constructed, the complete document of patient sequences was used to create embeddings with Word2Vec (Skip-Gram algorithm), using each patient sequence as a sentence, and each complete (untokenized) string label as a word. These pre-embeddings were then integrated in the model along with the original patient sequences.

By the time I had learned this, it was too late to use this information in the ConvAE model for this paper. I trained the ConvAE model on indices that were unique to each medical concept, but not meaningful in and of themselves. To understand a patient sequence as I constructed them, the indices would have to be mapped back to their codes (ICD9, LOINC, etc.). The consequence of this is that the model picks up purely on the co-occurrence of codes for patient sequences, much in the same way as in the Med2Vec model.

Using the labels as indices, pre-training word embeddings on these texts, and including these word embeddings in the model would have added an extra layer of meaning to each patient sequence. I would have liked to have known this earlier in order to see how it affected the performance of the model. However, this information is not clear from the original study or available in the study's code. In my opinion, the lack of this information in the original study makes the original study unreproducible.

I fed the created patient sequences into the model made available at the study's public code repository, with many changes and additions to the code that were necessary in order to get the model to train and evaluate to its finish.[8]

For the ConvAE model, I contributed the following:

- code to create a vocabulary from MIMIC-III's ICD9 diagnoses, lab events, and microbiology events, using unique integer indices mapped to medical codes.
- code to create patient sequences indexed to this vocabulary in the format required by ConvAE. Determining the correct format was only possible by careful analysis of the study and of the 200 synthetic patient sequences provided in the public repository.
- code for the equation for calculating the TF-IDF score for concepts, used to remove concepts that occurred too rarely to be useful to the model.
- code for evaluating the model from a checkpoint. The original code re-trained the model on the training data from scratch when evaluating the test data. Even my rented hardware was unable to accommodate this. This code proved to be a godsend when I had to rerun a second experiment with reduced vocabulary.
- code to run evaluation starting from the epoch and batch where it broke off from due to "broken pipe" instead of re-starting the whole process from the beginning. This was necessary on the first experiment due to its large number of events.
- code to deduplicate certain code sections, make it more readable, and make it easier to call from the command line with parameters.
- code to save predictions on the test set when evaluated on the best model learned on the training data.
- code to compare and evaluate the predictions to the original input, in order to compute the AUC-ROC score.
- code for calculating various statistics about the data and the results
- code for creating a mapping from the integer indices of the vocabulary to care-condition ICD9 codes from MIMIC-III.

The strengths of this model are in the density of information represented about a sequence of patient events. Using pre-trained word embeddings on the -descriptions- of events is a novel idea and I really would have liked to see the results of these representations on the prediction task. An additional strength lies n the fine granularity of the temporal aspect, which is ordered by time as well as date. The number and variety of different medical concepts - including, for example named entities gleaned from unstructured clinical notes - is an additional novel idea that plays exceptionally well with the use of word-embeddings.

A weakness of this model is the fact that, as it is presented in the original study, it is not reproducible. I had to reach out to the authors with questions about how the ideas they wrote about in the paper were reflected in the code they provided. The answers came so late that I could not integrate them into my experiment, unfortunately.

An additional weakness is that, unlike Med2Vec and Benchmark, the time aspects of the ordering are not divided into discrete temporal subsequences. Benchmark divides its events into discrete hourly periods, and Med2Vec divides its events into discrete visit sequences. In ConvAE, however, all events, though chronologically ordered, are merely placed one after another into one vector per patient. The model then divides these vectors into fixed LENGTH subsequences. It can therefore never really be certain (without extra coding and alterations to the model) how many temporal aspects are included in a subsequence.

In other words, does the 32 code-long subsequence include data for one day, for one hour, or for multiple hours? Into how many subsequences is one complete visit divided? I think it would be interesting to apply the temporal representation used in Benchmark or Med2Vec to the patient sequences of ConvAE. This would entail dividing the one long patient sequence per patient, into several discrete sequences per patient, perhaps by hour. These hourly subsequences could then be padded into fixed-lengths. This would provide an extremely fine-grained temporal aspect.

3.EXPERIMENTAL RESULTS

3.1.Benchmark Results

As discussed in the Benchmark study, results indicate that acute conditions seem to be easier to predict than chronic conditions, and that prevalence of conditions is not correlated with predictive ability. The Pearson correlation coefficient between condition prevalence and AUC-ROC score for my experiment is a mere -0.291 (with a p-value 0.16).

From examining Table 1, we can see that the Benchmark model has a robust, better than average ability to predict grouped diagnoses (care-condition) codes, independently of how common or rare they are within the data. The model infrastructure is difficult to get right, but once it is in place, the study is simple to reproduce in different contexts using the code they provided. The post-processed data is also reusable in different contexts, which makes this study and its code invaluable for exploring multiple methods of representing patients.

3.2.Med2Vec Results

Med2Vec, on the other hand, had very poor performance when predicting visit codes trained on MIMIC-III data. The predicted probabilities for diagnoses of a final visit were nearly all equally likely: each predicted probability was infinitesimally small with an insignificant range in variance (ranges from 0.00018 to 0.00022 for every predicted code probability was typical). This was the case for both predictions on full ICD9 codes (which the authors of the study did not recommend, due to the difficulty of the model to accomplish exact ICD9 prediction), and on the smaller subset of grouped 3-digit ICD9 codes.

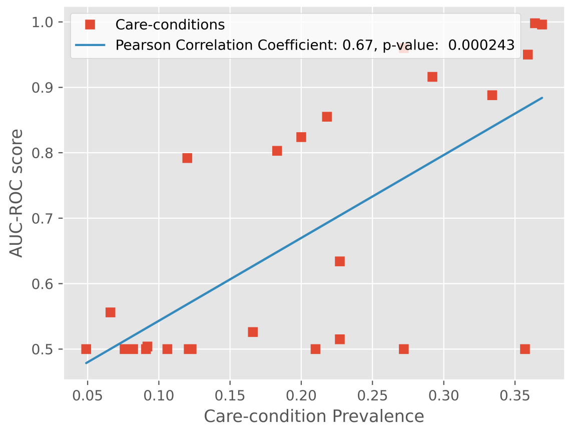
Because of this, I had to create a more lenient process for predicting AUC-ROC than just using the given predicted probabilities. Since the original study used top-30 Recall for measuring the ability of the model to predict diagnoses, what I did was to take the top true number of codes for a visit, and take that many top predicted codes for a visit. For example, if a last visit had 5 codes, I took 5 codes from the predictions with the highest probabilities. If any of these codes mapped to the grouped care-conditions, they were included as a true positive for that care-condition.

The results are presented in Table 1. Despite the fact that the prevalence of care-conditions in the training and testing set of both Benchmark and Med2Vec are similar, the AUC-ROC for

[8] https://github.com/landiisotta/convae_architecture

both experiments are quite different. In contrast to Benchmark, Med2Vec's ability to predict the presence of a care-condition is highly correlated with that condition's prevalence in the population data. The Pearson correlation coefficient is 0.67, with a p-value of 2.43e-4. In other words, the AUC-ROC scores that are higher by some care-conditions in Med2Vec are not the results of a superior predictive power of that model.

Figure 1



I believe that this is due to the fact that the amount of data from MIMIC-III that Med2Vec could use for training is insufficient for the richness of Med2Vec's feature space. I would go as far to say that Med2Vec is not an appropriate model for comparing benchmarks when trained on the MIMIC-III data.

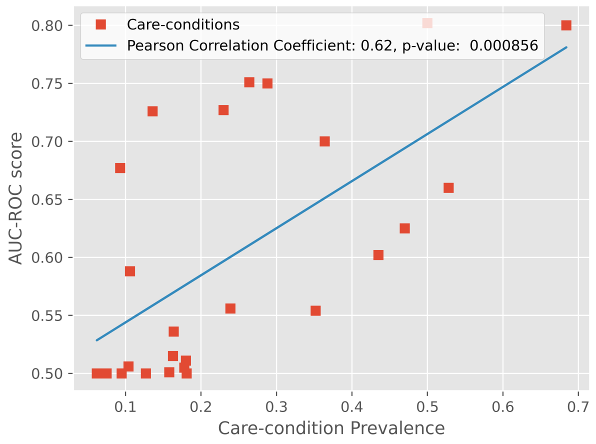
3.3.ConvAE Results

My first experiment with ConvAE, using diagnoses, lab and microbiology events, produced a model that did not predict a single care-condition for any patient in the test set, despite a care-condition prevalence distribution similar to Benchmark and Med2Vec. On careful inspection of the predicted codes, I noticed that they seemed to be fixated on lab events rather than diagnoses. I assumed that this is because lab events outnumber diagnoses per visit by a large margin, which caused me to try the experiment again, this time using only diagnoses. The new vocabulary had only about 1,000 concepts less after removing lab and microbiology events, but it had 100 times less the amount of events!

	Exp. 1 (diagnoses, lab, microbiology evens)	Exp. 2 (diagnoses only)
Nr. events	47.774.342	477.958
Nr. unique vocab	15.746	14.567

My assumptions based on the visual inspection of predicted codes were confirmed by this information: the number of lab events were so overwhelming, that the model basically ignored the diagnoses, which were insignificant in comparison.

The second model did much better in that it had usable results. This model's ability to predict care-conditions was only slightly better than that of Med2Vec. On a little more than half of the conditions, it had no ability to distinguish between classes. Additionally, like Med2Vec, its predictive performance was correlated with the prevalence of the condition, albeit with a slightly weaker correlation coefficient.



I interpret this to be a result of training the model on complete ICD9 codes. This is something which resulted in Med2Vec in a model that could not predict. The high number of codes, and their similarity, make it difficult to predict the exact code: it may be that a similar code was predicted instead. Using grouped 3-digit ICD9 codes in the vocabulary could have possibly led to better performance.

4.OPTIMIZATION

4.1.Benchmark Optimization

Due to financial constraints, the Benchmark models were only trained for five epochs. The training always broke off after one epoch, and I had to use checkpoints and manual updating of the epoch number in order to get it to finish training for five epochs. Thus, the performance of the model was limited to the improvement of the loss for this relatively small number of training rounds. Despite, this, the results are still reasonably good, which is an indication of the robustness of the model. To see how much results improve with further training would require more powerful hardware: an AWS image with more than one GPU, more parallelization and more computing power. This is expensive and I did not perform this optimization. This would definitely be an option, however, for researchers with more financial resources.

4.2.Med2Vec Optimization

Med2Vec can be optimized by training it on more data than was available in MIMIC-III after being filtered to exclude patients with less than two visits. The original study used about 100 times more data than what I had for training, while having only twice as many unique medical codes. Their average number of visits per patient were about twice as many as in the MIMIC-III data.

	Med2Vec original study		Med2Vec with MIMIC-III
	Dataset 1	Dataset 2	Total dataset
Patients	550.000	830.000	7.500
Visits	3.000.000	5.000.000	27.259
Avg. visits per patient	6,1	6,57	2,65

Because of the obvious problems caused by too few data, I did not continue to the next step of the experiment where I would include demographic data to be trained by the model. The original study also included other types of medical codes, such as medications and procedure codes. Including these features would optimize the model's performance - however, only with a much bigger dataset, equivalent to the size of the dataset of the original study. On the small dataset I had access to, this would have led to similar - or worse - results, such as I had when I trained the model on complete ICD9 codes instead of a subset of codes grouped into 3-digit ICD9s.

4.3.ConvAE Optimization

The original ConvAE study contained more types of medical codes than I used. Namely, in addition to ICD9 diagnoses and lab events (which I included in the first experiment, in addition to microbiology events), the study also used medications normalized to RxNorm, CPT-4 procedure codes, vital signs without their measurement values, categorical demographic information, and ICD10 codes mapped back to their ICD9 versions. Including these additional medical concepts in the patient sequences would have increased their ability to represent information.

However, as the results of my first experiment show, they would not have helped in the specific prediction task I was working on. In this way, more information does not improve the model performance when you are looking for a very specific outcome. In the case of my two experiments, the lab events overwhelmed the diagnoses, and the model very correctly focused on the large presence of lab codes. However, for the purpose of predicting (specifically) diagnoses, this is just noise. So running ConvAE on all of the types of medical codes used in the original study would have resulted in equally bad predictive performance (it could NOT have gotten worse)!

The study also mentions preprocessing clinical notes to extract medical concepts from free text to add to each patient sequence. Their initial medical concept vocabulary consisted of 57,464 concepts, of which they removed nearly half for having a score that fell beneath a threshold of 10e-6. Their final vocabulary count was 32,799. This is more than twice as many as in my model, which had (in the second experiment, with only diagnoses) 14,567 concepts. Thinking of the results of my two experiences, I'm not sure medical text entities would have improved the performance of the care-condition prediction task, unless these text were diagnosis-related.

Finally and most importantly, the authors of the original study integrated pre-trained Word2Vec embeddings of their medical concept vocabulary, using the patient sequences as sentences and the medical concepts as words. This additional layer of dense representation in training the model would make a very performant model for such a task as clustering the resulting dense embeddings in order to see how diagnoses are related to each other. However, in the specific predictive task I performed, it where I am looking for the presence or absence of specific grouped diagnoses, this may have been (much like the lab events) just more noise.

As was the case with Med2Vec, the ConvAE study had a much larger patient dataset than I did: nearly 1,5 million patients. Again, increasing the amount of patient data to train on would be the most optimal performance improvement.

5.DISCUSSION

Interpretability is much talked-about topic in AI for healthcare, and rightfully so. A less-discussed topic, however, is usability: in this case, the practical availability of the different proposed learned representations. It is important to compare the strengths and weaknesses of different models. However, without being able to put them into practice, we can't make use of the knowledge they provide. In this final section, I would like to discuss one important shortcoming shared by each of the discussed models.

Despite the publicly available code for Benchmarks, Med2Vec and ConvAE, setting up these experiments was incredibly time-consuming, agonizing and expensive. The CHARTEVENTS table of MIMIC-III does not fit in memory on an average laptop or even on a standard AWS EC2 instance. The image I ended up using was far beyond what was available in the "free tier". MIMIC-III (and the Benchmarks with their code) were made publicly available in order to lower barriers to research. However, the data-engineering knowledge required to take advantage of this goes far beyond the capabilities of most programmers and medical researchers. The financial resources necessary to support such hardware for long-running experiments is beyond the reach of most students, as well.

Med2Vec makes the code available for generating the weights and biases of visit and code representations directly from MIMIC-III data and has excellent documentation of how other data sources need to be formatted. The algorithms run quite smoothly (given that you either have the required hardware, or the know-how to set it up in the cloud). However, how to use these vectors in prediction tasks, once generated, is not well described in the paper in a way that translates easily to code. I was lucky to find other public repositories that I could use as a guideline for creating these functionalities. These other public repositories were not completely reliable either, as I had to rewrite important sections of the code from scratch, and do a lot of debugging. Getting value from Med2Vec required a lot of coding expertise.

ConvAE provides synthetic data, code to generate patient representations, and a very sparse explanation of how to run the provided code. It's clear that the data needs to be in a specific format, but this (unlike in the excellent section "Preparing the Data" in Med2Vec!) is not mentioned in the documentation. Using the model required a significant amount of coding for creating the vocabulary and patient sequences required for the model. Additionally, parts of the study (such as the exclusion of value for vital signs, categorality of demographics, integration of free-text clinical notes in the model and, most importantly, how to apply the aforementioned word-embeddings) were not explained in the source text or represented in the code. Trying to reproduce this model required significant effort and coding know-how, as well as contacting the authors.

When the barriers to using innovations in patient representation are too high, we can't make the kind of progress we need to in improving the quality of health care. Without the ability to quickly and easily use the learned representations presented in these various studies, the discoveries remain just an interesting academic exercise. With the experiments I presented in this paper and the accompanying publicly available code, I hope to have contributed to the practical knowledge necessary for making use of these models. I thus hope to make them more accessible to anyone interested in researching the possibilities of learned representations of electronic health records.

6.CONCLUSION

In this paper I have discussed the strengths and weaknesses of three learned patient representations using a phenotyping benchmark task. Results vary by dataset size and model reproducibility. Reproducing each model requires significant data-engineering, coding expertise and financial resources that may not be available to every researcher.

7.REFERENCES

1. Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Jimeng Sun. 2016. Multi-layer Representation Learning for Medical Concepts. arXiv:1602.05568

2. Harutyunyan, H., Khachatrian, H., Kale, D.C. *et al.* Multitask learning and benchmarking with clinical time series data. 2019. *Sci Data* **6,** 96.

3. Johnson, A., Pollard, T., Shen, L. *et al.* MIMIC-III, a freely accessible critical care database. *Sci Data* **3,** 160035. 2016.

4. Landi, I., Glicksberg, B.S., Lee, HC. *et al.* Deep representation learning of electronic health records to unlock patient stratification at scale. 2020. *npj Digit. Med.* **3,** 96.

5. Thomas Miklov, Kai Chen, Greg Corrado, Jeffery Dean. Efficient Estimation of Word Representations in Vector Space. 2013. arXiv:1301.3781v3

6. Miotto, R., Li, L., Kidd, B. *et al.* Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. 2016. *Sci Rep* **6,** 26094.

7. Narkhede, Sarang. Understanding AUC-ROC Curve. 2018. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

8. Ruan, T., Lei, L., Zhou, Y. *et al.* Representation learning for clinical time series prediction tasks in electronic health records. 2019. *BMC Med Inform Decis Mak* **19,** 259.

9. Sebastian Ruder. An overview of gradient descent optimization algorithms. 2017. arXiv:1609.04747v2

10. Yuqi Si, Jingcheng Du, Zhao Li, Xiaoqian Jiang, Timothy Miller, Fei Wang, W. Jim Zheng, Kirk Roberts, Deep representation learning of patient data from Electronic Health Records (EHR): A systematic review, Journal of Biomedical Informatics, Volume 115, 2021, 103671, ISSN 1532-0464.

11. Wei-Hung Weng, Peter Szolovits Representation Learning for Electronic Health Records. 2019. arXiv:1909.09248.

About the author:

Jennifer Koenig is a graduate student at the University of Illinois Urbana-Champaign where she is currently completing her Masters Degree in Data Science.

Code written by the author for this project is available on this GitHub repository.

The video presentation for this paper is available at

https://mediaspace.illinois.edu/edit/1_g57mnzot

Table 1: Selected care-conditions, their prevalence in each model population, and AUC-ROC score of model Cells in pink show elevated incidences of a condition in comparison to the other models.

		Benchmark Prevalence		Med2Vec Prevalence		ConvAE Prevalence		AUC-ROC		
Phenotype	**Type**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**	**Benchmark**	**Med2Vec**	**ConvAE**
Acute and unspecified renal failure	acute	0,216	0,212	0,292	0,297	0,264	0,268	0,799	0,916	0,751
Acute cerebrovascular disease	acute	0,075	0,066	0,049	0,065	0,093	0,103	0,911	0,500	0,677
Acute myocardial infarction	acute	0,103	0,108	0,082	0,071	0,136	0,130	0,725	0,500	0,726
Cardiac dysrhythmias	chronic	0,322	0,323	0,359	0,398	0,470	0,465	0,666	0,950	0,625
Chronic kidney disease	chronic	0,135	0,132	0,218	0,272	0,180	0,185	0,759	0,855	0,511
Chronic obstructive pulmonary disease and bronchiectasis	chronic	0,132	0,126	0,166	0,167	0,158	0,165	0,675	0,526	0,501
Complications of surgical procedures or medical care	acute	0,208	0,213	0,227	0,231	0,352	0,339	0,699	0,515	0,554
Conduction disorders	chronic	0,073	0,071	0,091	0,107	0,095	0,096	0,709	0,500	0,500
Congestive heart failure; nonhypertensive	acute	0,269	0,268	0,369	0,330	0,500	0,506	0,739	0,996	0,802
Coronary atherosclerosis and other heart disease	chronic	0,323	0,331	0,334	0,347	0,684	0,664	0,774	0,888	0,800
Diabetes mellitus with complications	chronic	0,095	0,094	0,120	0,112	0,181	0,172	0,862	0,792	0,500
Diabetes mellitus without complication	chronic	0,194	0,192	0,210	0,222	0,239	0,242	0,778	0,500	0,566
Disorders of lipid metabolism	chronic	0,291	0,289	0,272	0,349	0,364	0,367	0,715	0,960	0,700
Essential hypertension	chronic	0,421	0,423	0,364	0,380	0,528	0,524	0,661	0,988	0,660
Fluid and electrolyte disorders	acute	0,268	0,265	0,357	0,358	0,435	0,444	0,735	0,500	0,602
Gastrointestinal hemorrhage	acute	0,072	0,079	0,092	0,091	0,104	0,095	0,725	0,504	0,506
Hypertension with complications and secondary hypertension	chronic	0,133	0,130	0,200	0,229	0,163	0,167	0,739	0,824	0,515
Other liver diseases	acute	0,089	0,089	0,121	0,129	0,164	0,164	0,734	0,500	0,536
Other lower respiratory disease	acute	0,051	0,057	0,066	0,078	0,075	0,067	0,688	0,556	0,500
Other upper respiratory disease	acute	0,041	0,043	0,076	0,076	0,062	0,056	0,763	0,500	0,500
Pleurisy; pneumothorax; pulmonary collapse	acute	0,086	0,091	0,106	0,112	0,127	0,123	0,693	0,500	0,500
Pneumonia (except that caused by tuberculosis or sexually transmit	acute	0,140	0,135	0,183	0,173	0,178	0,183	0,799	0,803	0,505
Respiratory failure; insufficiency; arrest (adult)	acute	0,181	0,177	0,272	0,265	0,230	0,235	0,896	0,500	0,727
Septicemia (except in labor)	acute	0,143	0,139	0,227	0,205	0,288	0,300	0,830	0,634	0,750
Shock	acute	0,079	0,082	0,123	0,114	0,106	0,101	0,863	0,500	0,588