Joint Modeling of Electronic Health Records and Clinical Notes

Chirag Nagpal
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA
chiragn@cs.cmu.edu
https://github.com/chiragnagpal/747NN4NLP

Sai Krishna Rallabandi
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA
srallaba@andrew.cmu.edu
https://github.com/saikrishnarallabandi/747_assignment

Abstract

Significant progress has been made with modeling Electronic Health Records using Deep Learning Methods for Clinical Tasks. However, the text data present in the medical reports has not yet been leveraged comprehensively in such models. In this paper we propose to jointly model the medical profile of a patient using both the diagnosis codes as well as the textual reports accompanying them: a) by learning embeddings from natural language text in the form of notes by physicians and b) the Patients Historical Diagnoses and Procedures represented by ICD Codes. We further go onto exploit the learnt embeddings on predicting future clinical events and show the benefit of incorporating the textual information for better modeling the health profile of patients.

1 Introduction

Electronic Health Records (EHR) contain a coarse view of the medical profile of a patient. Depending on the system in use, hospitals record various variables including the patients' demographic information, all past histories of medical procedures performed, and diseases diagnosed. The availability of this longitudinal EHR data offers the possibility of deploying several machine learning and data mining techniques for medical data evaluation and prediction thereby revolutionizing medical informatics.

Deep Neural Models have made significant contributions to mining of such data, with various tasks being performed, including prediction of medical conditions and events which are encoded as ICD-9 codes in the subsequent admissions, predicting current conditions using Clinical Notes & Prediction of susceptibility to morbidity based on all prior data.

We observe that a significant amount of information is encoded in notes and reports corresponding to the patients including the doctors impression of any lab or radiology tests performed, any palliative treatments recommended if necessary etc. While there has been work to model this data using Neural Models, there has not been much research to glean from the notes and reports of patients alongside the ICD-9 Codes jointly using a single model. We propose to leverage this knowledge jointly with the patients past history in order to predict future admissions.

Our contributions in this paper can be summarised as follows

- Learn Embeddings for each patient from the ICD-9 Codes treating the Patients profile as a Language Model.
- Learn Embeddings from the Natural Language Text in Doctors Notes in each Patients Admission.
- Exploit the learnt embeddings to predict the future admission events jointly with prior patient history, using deep multimodal fusion.

2 Prior Work

Deep Learning has been applied extensively in the past to clinical tasks. Lipton et al. (2015) employed LSTM RNNs (Gers et al., 1999) to model continuous time domain signals like patient vital signs. One of the first such attempts to model EHR data using Recurrent Neural Networks was the Doctor AI System (Choi et al., 2016a). Doctor AI attempted to jointly predict the future ICD events along with time to next admission using Gate Recurrent Units (Graves et al., 2009). Another work of the same author (Choi et al., 2016b)

attempts to learn embeddings from the ICD-9 information that includes the Medication, Procedure and Diagnostic Codes for which they employ Skipgrams (Mikolov et al., 2013) along with ReLU activations (Nair and Hinton, 2010).

3 Dataset

We use the MIMIC-III dataset (Johnson et al., 2016), which stands for 'Medical Information Mart for Intensive Care'. The Dataset consists of vital signs, medications, laboratory measurements, observations and notes charted by care providers, fluid balance, procedure codes, diagnostic codes, imaging reports, hospital length of stay, survival data of over 38,000 Patients aggregated over corresponding to over 50,000 distinct admissions aggregated over a period of 11 years. Being a one of the larger and publically available dataset, it is the most popular for clinical informatics tasks.

4 Clinical Tasks

We want to empirically validate if it is possible to predict the Diagnostic Codes, given just data from the patient health records. This has significant clinical impact, certain rare and harder to diagnose diseases have a tendency to be under-reported. Using the available information, We define three predictive clinical tasks for related, cardio-circulatory conditions which are listed below, along with there corresponding ICD-9 codes in Table 1

	ICD CODE	DISEASE
TASK-EH	401	Essential Hypertension
TASK-HF	428	Heart Failure
TASK-HA	427	Cardiac Arrhythmia

Table 1: Clinical Tasks

5 Embedding Clinical Reports

The clinical reports are broadly grouped into 16 categories, and consist mostly of some natural language text recorded by the medical practitioner, that includes multiple medical concepts and some metadata about the patient, including admission units, serial numbers, name of Caregivers involved, Dates.

We first build regular expressions to strip all the metadata from the clinical reports inorder to learn embeddings. We then proceed to utilise, Continuous Bag of Words (CBOW) and Skipgram (SG) models to learn the per word embeddings. We observed, and corroborated from previous research (Ayyar, 2017) that since these reports were hand generated, there were significant instances of misspellings, and thus the use of Character Level, or Subword Level Embeddings may perform better, and be robust to these idiosyncarcies of the data.

For both CBOW & SG we used a context window of 5 and extracted embeddings of dimensionality 128 & 256. Since each admission has multiple reports that correspond to 16 different tests, we aggregate the word vector representations for each individual test by averaging over them. These averaged vectors for each test are then concatenated together to represent the patients stay in a continuous space.

Table 2 & Figure 1 represents the performance of a Logistic Regression Classifier trained on the various different embeddings we extract in terms of Area Under the Receiver Operator Curve (AUROC). We observe that Skipgrams with a dimensionality of 256 outperformed all other representations, although the improvement was only of a small margin.

EMBEDDING	TASK-EH	TASK-HF	TASK-HA
CBOW-128	0.6673	0.7885	0.7867
CBOW-256	0.6685	0.7856	0.7871
SG-128	0.6756	0.7970	0.7991
SG-256	**0.6803**	**0.7999**	**0.8006**

Table 2: AU-ROC for Various Embeddings

5.1 Experimental Protocol

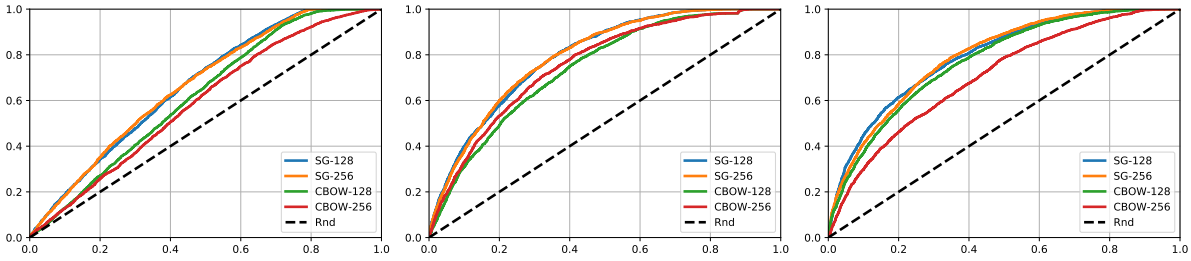
We proceed to utilize the previously learnt embeddings in order to train a classifier in a supervised fashion, thus for each admission corresponding to a patient, we aim to predict if the patient would be diagnosed with one of the described tasks at the end of there current admission. We perform training on an 80% of the admissions and test on a held out set of 20% of the admissions. We create the splits in a patient independent fashion, such that no single patient lands in both the splits.

5.2 Model Learning

In this section we describe the Baseline Models, along with the Deep Neural Models that we trained.

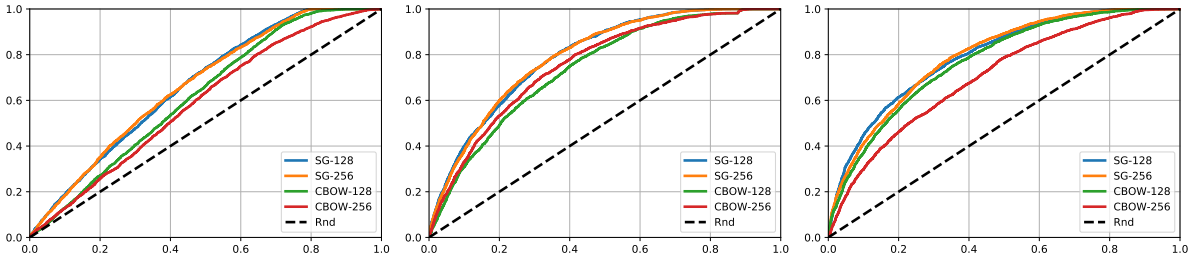
5.2.1 Baselines

- Logistic Regression With $\ell 2$ Regularisation



TASK-EH TASK-HF TASK-HA

Figure 1: ROC Plots of LR Model trained on Different Learnt Representations



TASK-EH TASK-HF TASK-HA

Figure 2: ROC Plots of the Various Models trained on SG-256

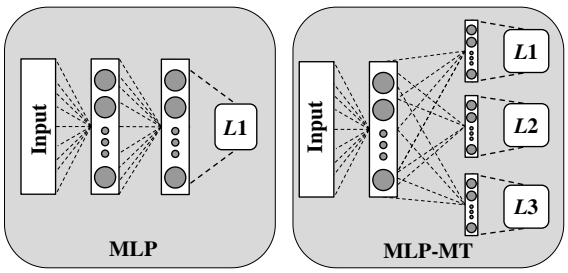


Figure 3: Standard MLP vs. MT-MLP

Logistic Regression is a simple baseline, which is useful as a diagnostic tool for determining the hardness of the Learning Task. We apply an $\ell 2$ penalty on the weights vector with α, the regularisation parameter set to 10^{-4}

- Random Forest Ensemble

  We use a Random Forest Estimator with 100 trees, with Gini Index as the criteria for splitting.

5.2.2 Neural Models

We describe the Multilayer Perceptron Models we train in the following section. Figure 3 presents a representative view of the neural models described.

- Multi-Layer Perceptron (MLP)

  The MLP model consists of a Single Linear Layer with Sigmoid activation along with a final Linear layer trained to optimize cross entropy loss.

- Multi-Task Multilayer Perceptron (MT-MLP)

  The multitask MLP is a variation of the MLP model, it consists of A Single Linear Layer with Sigmoid Activations of the same dimensionality as the input. For each task, we then add final layer, each of which is optimized for Cross Entropy Loss. Finally the Cross Entropy is aggregated for each task. We hypothsize that since the three tasks are related conditions, the multitask model will be able to generalise better by learning a more robust intermediate representation.

We utilise the PyTorch Framework to implement the Neural Models as described above, and train using Adam (Kingma and Ba, 2014) with a Learning Rate of 10^{-3} for 3000 epochs on the Training Set. Table 3 and 2 presentes the AU-ROC and ROC Curves for the Test Dataset. We observe that MLP-MT outperforms MLP for each class, it is also worth mentioning that since the MLP-MT model is trained jointly and shares parameters, it has lesser overall number of parameters, and thus trains much faster.

MODEL	TASK-EH	TASK-HF	TASK-HA
RF	0.6596	0.7695	0.7822
LR	0.6709	0.7959	0.7948
MLP	0.6812	0.7992	0.8036
MLP-MT	**0.6818**	**0.8002**	**0.8037**

Table 3: AU-ROC for Models on SG-256

6 Predicting diagnosis codes using the patient visit data

In this section, we describe the systems we built to predict the diagnosis codes of patients given the existing visit data. We first describe the correspondence between the health care data and natural language. We then describe baseline approaches that try to model and predict the clinical diagnosis codes using sequence to sequence models. We show a procedure to incorporate the embeddings learnt in the previous sections in the prediction models. We conclude by discussing the scope of the project and the subsequent approaches that we are going to work on.

6.1 Correspondence between Clinical data and natural language

The sequentiality of ICD codes in a diagnosis points out a close analogy between them and natural languages. Specifically, we can view the sequence of ICD codes of one admission as a sentence and each ICD code as a word in a natural language. This leads us to apply the techniques typically used in NLP such as continuous bag-of-words (CBOW), etc. Diving even further, given that typically the diseases follow a hierarchical structure, we can view these structures similar to the parse trees in NLP. In addition, there might exist 'health grammars' similar to syntactical grammars in NLP. In this case, a health grammars might refer to (latent) deterministic biological and environmental patterns(variables) that model and dictate the progression of one's health condition over time. A progression from Blood Pressure to hypertension might serve as an example of the same. It might be useful in such cases to model such correlations and predict them.

6.2 Prediction Tasks on Clinical data

Having established the connection between words in natural languages and clinical code distributions, we discuss and apply in this subsection the techniques used in NLP to predict the diagnosis codes per visit. We have outlined the tasks of interest in the table 4. We begin the discussion with a method to predict all the codes in a given visit. This would enable us to find if there is some correlation between the type of diseases that 'co occur'. This task is similar to language modeling in NLP and we use a baseline model inspired by RNNLM for this. We then move on to the more interesting task: Predicting the diagnosis codes of visit at time t+1 given the diagnosis codes of visit at time t. A model that can perform this task is particularly useful as it can act as a preventive mechanism. This task is specifically similar to the task of statistical machine translation if we consider the codes corresponding to visit at time t as the source and those of visit at time t+1 as the target language. It is interesting to note here that there might be multiple visits (lets say, N) by a single patient, in which case we move away from the analogy. We have also built a baseline system to predict the most important diagnosis code in a visit. All the models were built using Dynet (Neubig et al., 2017) and we have included the DIAGNOSIS.csv file from the dataset in the repository provided for reproducibility of the code.

Table 4: Prediction Tasks

Task	Description
Task A	Is there a structure in output codes?
Task B	Predict diagnosis for next visit
Task C	Predict most important code in next visit
Task C_1	Predict using first N codes

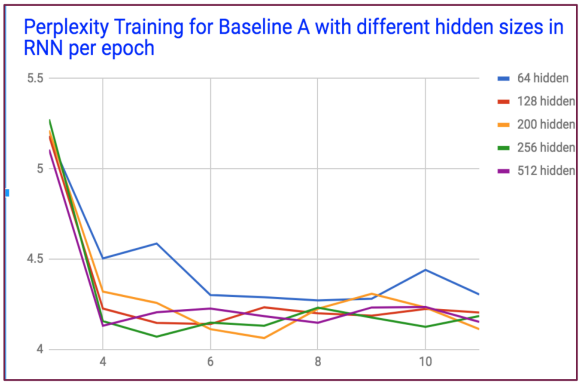


Figure 4: Training per epoch of Baseline A

6.3 Baseline for Task A: Predicting all the diagnosis codes in a particular visit

We have first extracted the clinical codes corresponding to each visit irrespective of the patient

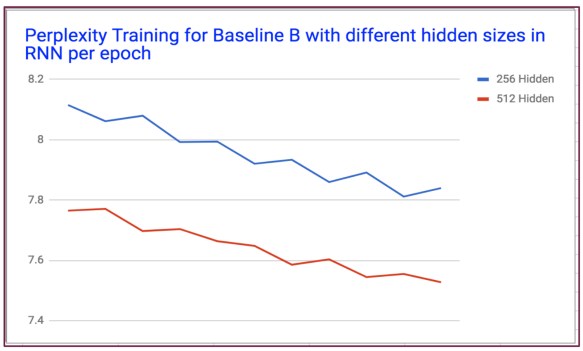


Figure 5: Training per epoch of Baseline B

identity. There were 64314 sequences in total and we have trained to predict the same sequence using a unidirectional RNN in a language model fashion in the sequence of the clinical codes. The ICD codes in the diagnosis are listed in the order of their priority. Therefore, we felt that it is important to consider the order while training the model. We have used 128 neurons as the hidden representation and 64 dimensional vectors as embeddings for the ICD codes. Learning rate was 0.01 and SGD was used as the optimizer without dropout. There was no batching performed. The performance of the model was evaluated based on the predictions for all the codes and the order as well. As this is usually the case also with language modeling, we have used the measure typically used in LM: perplexity to optimize and measure the performance of the model. In addition, we have also considered the option of using a sorted code error rate for tuning the hyper parameters with the intention to at least predict all the diagnosis codes even if the ordering was missed by the model.

6.4 Baseline for Task B: Predicting the diagnosis codes for the next visit

This is a model that predicts the clinical codes corresponding to the next visit using the clinical codes corresponding to the current visit. We have used a sequence to sequence architecture using two RNNs : one for encoding the codes corresponding to the current visit and another for decoding the codes for the next visit conditioned on the encoder's output. We have used the same configuration for RNNs as was described in the subsection before. However, the losses back propagated via this model only include those in the decoder. We have not used any attention mechanism in this baseline.

Table 5: Performance evaluation of systems to predict diagnostic codes. The joint model has only been evaluated in task B.

Config	Train Ppl	Validation Ppl
Baseline Task A	5.73	6.24
Baseline Task B	7.32	8.81
Baseline Task C	0.42	0.48
Baseline Task C (N=5)	0.44	0.57
Joint Model (Task B)	**7.22**	**8.63**

6.5 Baseline for Task C: Predicting the most significant code in the next visit

This is a model that is trained to predict the most significant disease code in the next visit. The training architecture and the parameter space for this baseline are equivalent to the previous case except that the output now is not a sequence but a single code. We have tried using only N codes for the encoder, varying N from 1 through 5 to see if that plays a role in the prediction. In this case, the evaluation metric is accuracy of the predicted class. However, for comparison, we still refer to it as perplexity in the table 5.

6.6 Joint Model: Incorporating the learnt Embeddings into the model training

All the baseline models have been trained only using the medical codes. However, the handwritten reports mentioned in sections 4 play a significant role in understanding the status of the patient. We hypothesize that incorporating the details of the report can enhance the training of the model. Along these lines, we train this model which demonstrates one possible way to incorporate the report embeddings learnt in the earlier section into the training procedure. We have used the report embeddings as the initial state for the RNN while training on the medical codes for this. Other methods of training such as adding the embeddings as additional features, etc will be a subject for further exploration.

7 Conclusion and Scope of the project

We hshow that the textual information in the medical reports can assist in the process of predictive analysis for clinical tasks. We plan to explore different joint training strategies for the Check Point 2. In the current systems we have used a nominal variant of sentence level embedding obtained by a linear combination of the word level embed-

dings within. In addition, the training of these embeddings was carried out independent to the final model that predicts the medical codes. We plan to look at

- Better modeling strategies at the sentence and document level
- Techniques to jointly train both the textual as well as medical codes
- Using structured models such as Tree LSTMs for the input modalities, and exploring graphical regularisation techniques for the output space predictions as described in (Nagpal et al., 2017)

Acknowledgments

We would like to thank the instructor, Prof. Graham Neubig and all other TAs for reviewing the document. We would also like to express gratitude to Prof. Artur Dubrawski for access to the MIMIC-III dataset and computing resources.

References

Sandeep Ayyar. 2017. Tagging patient notes with icd-9 codes .

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016a. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*. pages 301–318.

Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. 2016b. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pages 1495–1504.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with lstm .

Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. 2009. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence* 31(5):855–868.

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

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Zachary C Lipton, David C Kale, and Randall C Wetzell. 2015. Phenotyping of clinical time series with lstm recurrent neural networks. *arXiv preprint arXiv:1510.07641* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Chirag Nagpal, Kyle Miller, Tiffany Pellathy, Marilyn Hravnak, Gilles Clermont, Michael Pinsky, and Artur Dubrawski. 2017. Semi-supervised prediction of comorbid rare conditions using medical claims data .

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. pages 807–814.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980* .