Deep Learning on Clinical Notes

Adam Lieberman, Ravish Chawla, & Garrett Mallory

***Abstract*— There are over 5,500 registered hospitals in the United States. When a patient visits a hospital, clinical notes are taken based off on patient symptoms, lab results, and surgical notes. These notes help doctors actively track and diagnose their patients. However, clinical notes vary from hospital to hospital and interoperability standards are not preserved in the sharing of these electronic health records. Additionally, clinical records can be messy from a data stand point and be difficult to interpret. Thus, there are often incorrect diagnoses. This project aims to leverage the power of the MIMIC III database, deep learning, and big data technologies to build a system to assist doctors in quickly and accurately diagnosing patients.**

I. OVERVIEW

In this paper, we detail the construction of Dr. ANN, our web-based deep learning system. The system inputs text data as represented by a clinical note and outputs a diagnosis for that clinical note in the form of an ICD-9 code. Additionally, keywords in the clinical note are extracted, highlighting which words are most relevant for the predicted diagnosis. This system delivers a few main advantages:

1) The system can serve as a second opinion for a doctor who has a diagnosis in mind.
2) Our system can review hundreds of clinical notes and diagnose these patients before a doctor can review and diagnose a single patient.
3) New data can constantly become training data which allows the system to continuously update and achieve more accurate diagnoses.
4) Key concepts relating to the predicted diagnosis from a clinical note query can be extracted to help the user better understand and justify the model output.

Currently, there has been a great deal of research on using deep learning, particularly using recurrent neural networks (RNNs) to diagnose patients, yet we have found no system has been deployed in a hospital or doctor's office. This is partly due to machine error. Computers are not perfect and do make mistakes. Diagnosing a patient can be a life-changing event and to put this fate solely in a computer's hands can be quite dangerous. However, a computer trained to recommend diagnoses like a doctor can serve as a second opinion. The deep learning algorithm could potentially discover diagnoses the doctor never considered and could serve as a great point of reference to assist the doctor in diagnosing patients. We explore this ongoing effort through the following model pipeline:

1) Create a term frequency-inverse document frequency (tf-idf) feature vector from the clinical note .
2) Use a conditional random field (CRF) on the clinical note to extract the most relevant words from the note relating to the diagnosis.
3) Input the tf-idf feature vector into a long short-term memory (LSTM) network to obtain a diagnosis code.

The web based system and application for clinical note prediction can be found at www.doctorann.xyz and a video presentation showcasing the system with a demonstration of the web application can be found and viewed at https://www.youtube.com/watch?v=zNiDHWZhEOo.

II. DATA

We will be utilizing data from MIMIC-III, a large publicly available database comprising of de-identified health related data associated with approximately 60,000 patients from the critical care units of Beth Israel Deaconess Medical Center over the time period from 2001 to 2012. The database is comprised of static and dynamic data ranging from birth dates to demographics to laboratory test results to diagnoses to imaging reports to clinical notes. For our purposes, we will be analyzing clinical notes and predicting associated diagnoses. Thus, we will particularly make use of the following tables:

- DIAGNOSES_ICD - contains patient diagnoses in the form of an ICD-9 code
- NOTEEVENTS - contains the following types of clinical notes:
  - Discharge summary - A summary written by the physician (or possibly a team of physicians) at the time of discharge from the ICU.
  - Radiology reports - Reports from imaging procedures such as MRI, CT, and Ultrasounds.
  - Nursing progress reports - Daily notes from the nursing staff.

III. ARCHITECTURE

A. Technology Stack

Our application is based on the traditional server-client model with a client hosting a front-end application for users to interact with and the server enabling backend preprocessing and computations on the dataset. The server consists of many different technologies. The main services we use to host our Python 2.7 code are Flask and Heroku. With Python, we use several libraries and frameworks:

- TensorFlow - A deep learning framework that enables us to build scalable neural networks
- Keras - A high-level wrapper for TensorFlow which allows us to create deep learning models in a more transparent manner

- NumPy - A library that allows us to perform linear algebra in a less computationally expensive, vectorized fashion.
- SciPy - A scientific library that provides us sparse data structures
- Pandas - A library that allows us to store data in a clean and less computationally expensive dataframe structure
- CliNER - A natural language processing library similar to CTakes that allows feature extraction from clinical notes

Our web application is built with HTML, CSS, JavaScript and JQuery. We use the Flask microframework and Jinja2 to bridge the frontend and backend together. Both services will be hosted separately on Heroku. The user will interact with the client application, which will communicate with the backend service to obtain the results of the user's query and provide them on the frontend for visual display.

Big-data scale preprocessing and deep learning is computationally expensive and thus we make use of Microsoft Azure's memory-optimized and GPU enabled machines to speed up our development time. We use a D15 v2 machine with 20 cores and 140 GiB RAM to run our preprocessing. For the LSTM model training, we use a NC12 machine with 12 cores, 112 GiB RAM and 2x K80 GPUs.

B. Preprocessing

Our MIMIC III clinical notes are originally very messy and hardly machine readable. There are misformatted and irrelevant dates and non-alphanumeric characters like asterisks, dashes, slashes, and special characters such as newline characters. Additionally, the notes contain uneven whitespace. Figure 1 shows what a clinical note may look like. It is synthetically generated and not directly from the MIMIC database.

Our first step in preprocessing this data is to remove the dates. These dates do not contribute as clinical terms so they are not needed. Next, we remove the non-alphanumeric characters, special codes, and extraneous white space. Lastly, we lowercase all characters. Our cleaned text sample is shown in figure 2.

We see that the above cleaned sample is much more machine readable and is better suited to construct features from.

Additionally, our ICD9 codes need reformatting. We will be using broader ICD9 codes instead of the longer form to allocate more rows in the train and test set for each label. Thus, we will be predicting all categories of the ICD9 code, up to the decimal point. For instance, we will predict a diabetes diagnosis as 250 instead of predicting type 1 diabetes as 250.01 or type 2 diabetes as 250.02. In our case, 250.01 and 250.02 will both become mapped to 250. This allows us to more broadly classify patient's symptoms. There are three cases where we need to make modification to our codes:

- "V" - If the ICD-9 code present in the diagnosis table begins with "V" then we return the first 3 characters in the present ICD-9 code.

Discharge Instructions: General Instructions ?????? Have a friend/family member check your incision daily for signs of infection.
?????? Take your pain medicine as prescribed.
?????? Exercise should be limited to walking; no lifting, straining, or excessive bending.
?????? You may wash your hair only after sutures and/or staples have been removed.
?????? Unless directed by your doctor, do not take any anti-inflammatory medicines such as Motrin, Aspirin, Advil, and Ibuprofen etc.
?????? Clearance to drive and return to work will be addressed at your post-operative office visit.
CALL YOUR SURGEON IMMEDIATELY IF YOU EXPERIENCE ANY OF THE FOLLOWING
?????? New onset of tremors or seizures.
?????? Any confusion or change in mental status.
?????? Any numbness, tingling, weakness in your extremities. Followup Instructions:
You will need to see the nurse practitioner 14 days post-operatively for suture removal. Please call [**Telephone/Fax (1) 1669**]
for the appointment.
You will need to follow up with Dr. [**Last Name (STitle) **] in 4 weeks with a Head CT of the brain.
Completed by:[**2118-12-9**]" 184,28063,121936,2125-02-16,,,"Discharge summary","Report",,,"Admission Date: [**2125-2-9**] Discharge Date: [**2125-2-16**]

Fig. 1. Original Clinical Note Input Sample

discharge instructions general instructions friend family member check incision daily signs infection. take pain medicine prescribed. exercise limited walking lifting straining excessive bending. may wash hair sutures staples removed. unless directed doctor take anti inflammatory medicines motrin aspirin advil ibuprofen etc. clearance drive return work addressed post operative office visit. call surgeon immediately experience following new onset tremors seizures. confusion change mental status. numbness tingling weakness extremities. followup instructions need see nurse practitioner 14 days post operatively suture removal. please call telephone fax 1 1669 appointment. need follow dr. last name stitle 4 weeks head ct brain. completed

Fig. 2. Cleaned Clinical Note Sample

- ”E” - If the ICD-9 code present in the diagnosis table begins with ”E” then we return the first 4 characters in the present ICD-9 code.
- Else - If the ICD-9 code does not begins with either ”V” or ”E” then we return the first 3 characters in the present ICD-9 code.

After obtaining the truncated ICD-9 classification codes, we join them on our clinical notes from the hospital admission ID (HADM_ID). This is the primary key common to both datasets. After performing the join the dataset takes the following form:

HADM_ID	Note	ICD_9
12	Text 1	V87
12	Text 2	250
12	Text 3	340
15	Text 4	589
15	Text 5	589
15	Text 6	560
20	Text 7	250

TABLE I

SAMPLE TABLE JOIN ON HADM_ID

Above, each hospital admission ID has a note with a corresponding ICD-9 code. Each hospital admission corresponds to a specific patient in the hospital for a specific hospital stay. This means that each hospital admission ID corresponds to one patient’s entire stay in the hospital. Over the course of this visit, the patient could have multiple clinical notes and ICD-9 diagnosis codes prescribed to them. With the hospital admission ID we can match a corresponding ICD-9 code to a patient’s clinical note. In Table 1, we see that there is a many-to-many relationship between the set of notes for a patient and the set of ICD-9 codes. Thus we create a set from the list of ICD-9 codes associated with each patient’s hospital visit and perform an inner join for it with the notes table. Our synthetic example from Table 1 now looks as follows:

Note	ICD_9
Text 1	[V87, 250, 340]
Text 2	[V87, 250, 340]
Text 3	[V87, 250, 340]
Text 4	[589, 560]
Text 5	[589, 560]
Text 6	[589, 560]
Text 7	[250]

TABLE II

SAMPLE TABLE ICD-9 SET INNER JOIN

After formatting our data in this manner, we have over 2 million samples of clincal notes paired with potential ICD-9 codes. With our data structered in this manner we can observe that the distribution is skewed left with an average count of 3,000.

C. Feature Construction

CliNER, an open-source natural language processing system for named entity recognition in clinical text of electronic health records, offers an out-of-the box silver model trained on MIMIC data. This model identifies clinical concepts from text data. We can pass in 40,000 clinical note into this model to extract clinical concepts and phrases, which allows us to build a repository of the medical terminology present in our clinical notes data. After our medical repository is built, we need to clean it. We remove stop words from all clinical phrases then pass it through a scikit-learn module to generate permutations of the subphrases in our phrases with all n-grams in the range 1-4. We take all permutations that appear in at least 2 documents to make reduce largely irrelevant concepts. Without the minimum document constraint, we have a vocab size of 57,031 vocabulary, and with it we have a size of 15,651. Multiplied out by our 2M notes, we found that our computational resources were insufficient to handle the 118.8 GiB file generated by the larger vocabulary size. The small size generates a much more computationally tractable in-memory size of 32.6 GiB.

With a present vocabulary, we can create tf-idf features for our set of clinical notes. Tf-idf allows us to reflect how important a phrase is to a document in a corpus. Some words are very frequent like ”the”, ”a”, ”is”, but carry very little meaningful information about the actual content of the document. If we were to look at a frequency count in a document that has many of these non-meaningful phrases our model might perform worse as it is not giving importance to the rarer yet more interesting terms.

Using an n-gram of 1, for example, we find that the top 15 words in our vocabulary are as follows:

Vocabulary Word	Number of Occurrences
pt	2987186
left	2179431
name	2044487
right	1997474
mg	1941351
ml	1891773
patient	1686424
last	1556429
chest	1287864
plan	1261318
normal	1258367
reason	1198831
clip	1156675
pain	1136394

TABLE III

TOP 15 WORDS FROM VOCABULARY

In comparison, the table below contains the bottom 15 words from our vocabulary with their respective counts:

Vocabulary Word	Number of Occurrences
codac	1
holdover	1
highlighter	1
dermatography	1
amish	1
surgicals	1
fiexible	1
protoptic	1
nucleoplasm	1
carpial	1
racer	1
dynamical	1
fajitas	1
particiation	1

TABLE IV
BOTTOM 15 WORDS FROM VOCABULARY

We see that our top words are fairly common clinical terms, areas of the body, and conditions. The bottom 15 words seem to be misspellings and uncommon words that one would not find in a clinical note.

D. Dimensionality Reduction

Having obtained our feature vectors, we note that they are of very high dimension: 57,031 as our vocabulary is of that size. To further reduce the size of our feature vector we apply singular value decomposition (SVD) and take the 1000 most relevant features.

SVD is the factorization of a real or complex matrix. It is the generalization of the eigendecomposition of a positive semidefinite normal matrix to any m x n matrix via an extension of polar decomposition. The SVD of a real or complex m x n matrix M is a factorization of the form $U\Sigma V^*$ where U is an m x m real or complex unitary matrix, Σ is an m x n rectangular diagonal matrix with non-negative real numbers on the diagonal, and V is an n x n real or complex unitary matrix. The diagonal elements $\sigma_i \in \Sigma$ are the singular values of M.

To create our SVD model we use sklearn and instantiate an SVD object with the number of components set to 1000, we then fit_transform our training data and obtain U. We can then obtain the variance ratio, Σ and perform a dot product between U and Σ to arrived at the reduced dimensionality representation of our tf-idf feature vectors used to train our model.

E. Generating Labels

With our feature vectors present, we need to generate our labels corresponding to each feature vector. To do so we create variable K-Hot encoded vectors for each feature vector. Each vector for a hospital admission will consist of a length N vector with K values set to 1, where N is the total number of ICD-9 codes in our multi-class classification and K is the total number of ICD-9 codes associated with a person's hospital visit. In our case $N = 1070$ as we have 1070 distinct ICD-9 codes. Each of the 1070 column's represents an ICD-9 code. So, if there is a 1 in that column than that means the clinical note matched with that specific ICD-9 code. The K value could be different for each person.

In Table II we see that for the text1 sample $K = 3$, but for the text7 sample $K = 1$. This means that the feature vector corresponding to note 1 would have 3 1's in corresponding columns for ICD-9 codes V87, 250, and 340. The rest of the entries would be 0. In text7 there would only be one 1 in the corresponding column for ICD-9 code 250. The rest of the entries would be zeros.

To create the label vectors we first create a dictionary where each key is an ICD-9 code and each value is a number 0 through 1069. We create a vector of zeros that is of size length of the distinct ICD-9 codes. For each code in our set of ICD-9 codes we look up the index in the dictionary and set the vector of that index equal to 1.

F. Conditional Random Fields

Conditional Random Fields (CRFs) are a statistical sequential modeling method frequently used as an alternative to Hidden Markov Models (HMMs) in fields like bioinformatics as they can model a much richer set of label distributions, define a much larger set of features, and have arbitrary weights. In a CRF, each feature function is a function that takes in:

- A sentence, S
- The position i of a word in S
- The label l_i of the current word
- The label l_{i-1} of the previous word

The CRF outputs a real valued number. Each feature function f_i is assigned a weight λ_i. Given a sentence S we can score a labeling l of S by summing the weighted features over all words in S. Our score can be computed as follows:

$$score\,(l|s) = \sum_{j=1}^{m}\sum_{i=1}^{n} \lambda_j f_j(S, i, l_i, l_{i-1})$$

We can then take the scores and transform them into probabilities $p(l|s)$:

$$p(l|s) = \frac{exp\,[score(l|s)]}{\sum_{l'} exp\,[score(l'|s)]}$$

$$= \frac{exp\left[\sum_{j=1}^{m}\sum_{i=1}^{n} \lambda_j f_j(S, i, l_i, l_{i-1})\right]}{\sum_{l'} exp\left[\sum_{j=1}^{m}\sum_{i=1}^{n} \lambda_j f_j(S, i, l'_i, l'_{i-1})\right]}$$

The CliNER module has a pre-trained CRF model using MIMIC data. We make use of this pre-trained CRF model to pass in a clinical note and identify the key clinical problems and treatment categories of terms within this note. For instance, we can pass the following clinical note into our conditional random field model:

> Patient has been suffering from headaches and a sore throat and has been prescribed Advil.

Fig. 3. Sample Sentence for CRF

and obtain the following patient problems and treatments labeling:

Patient has been suffering from <problem> headaches <problem> and <problem> a sore throat <problem> and has been prescribed <treatment> Advil <treatment>.

Fig. 4. CRF Clinical Sample Sentence Labeling

We see that the model has been able to label headaches and a sore throat as a problem and has identified Advil as the treatment.

G. Long Short-Term Memory

Long short-term memory networks are a special type of recurrent neural network capable of learning long-term dependencies. Recurrent neural networks have loops which allow the network to use information from the previous passes, acting as finite memory. Here the older the information, the less usable it is. An LSTM is able to actively maintain self-connecting loops with memory output to prevent memory degrading. The LSTM is trained to select what gets passed into this memory output. With this, the networks does not have to worry about new information degrading the prior information.

Using TensorFlow and Keras we create a sequential model with an LSTM layer consisting of 32 neurons connected to a dropout layer with a rate of 0.5, which is then conencted to a dense layer. The feature vector will pass through these layers and probabilities for each ICD9 code will be generated.

H. User Interaction

Once our deep learning algorithm has been finely tuned, it is ready for predictions. To make our system user friendly, we are creating a Python-backed web application where doctors and patients can input their clinical record and view the model's predicted diagnoses along with the words most responsible for the diagnosis. In our system, they will find information regarding the top 3 predicted diagnoses and sample clinical reports for each predicted diagnosis. This will allow them to evaluate and gain insight into the condition. We host the system on the Heroku platform at the following link: https://doctorann.xyz.

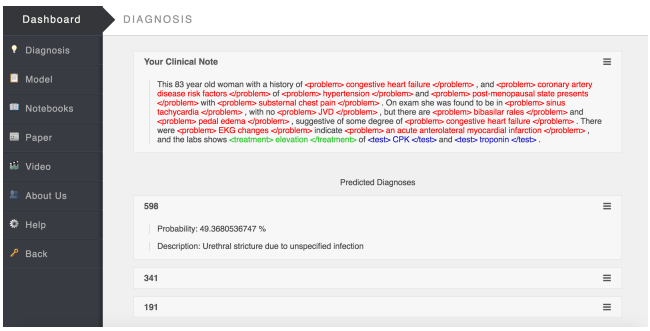

Fig. 5. Dr. ANN Dashboard

I. Visual Pipeline

The entire pipeline of our system is as follows:

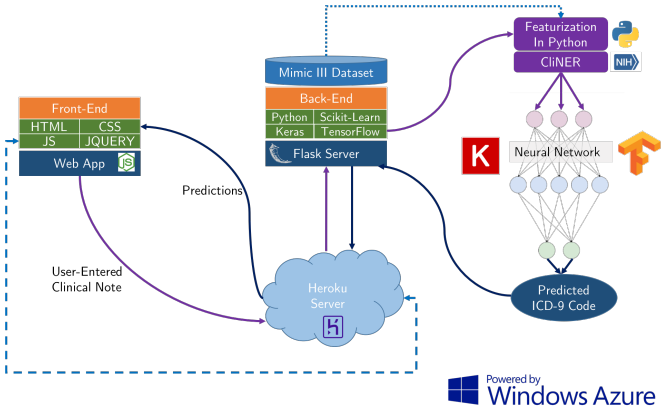

Fig. 6. Dr. ANN system pipeline

This architecture diagram details the use of our open source technologies, how the Heroku serve serves as a platform middleman between the front-end and back-end, and how the neural network is situated in the information pipeline.

Finally, all of this is powered by cloud services hosted on Windows Azure. The use of cloud services was critical to enable storage and processing of data that would have been unapproachable on personal computing machines.

IV. Performance Measurements

There are several types of sample-based metrics that we can use in multi-label classification to measure performance. To define our metrics, we will let Y_i denote the set of predicted labels, Z_i denote the ground truth labels, and n to denote the number of samples. We will use the following metrics with their respective definitions:

$$Accuracy = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|}$$

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Z_i}$$

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Z_i|}{|Y_i|}$$

$$F1 = \frac{1}{n}\sum_{i=1}^{n}\frac{2|Y_i \cap Z_i|}{|Y_i| + |Z_i|}$$

V. Results

Below we see a confusion matrix of the results of our LSTM model. This was created by predicting ICD-9 codes for each of our 300k test samples from a train-test split on our input data. We take the top k most likely predictions where k is the number of actual predictions that exist in sample. Next we flattening the length 1070 prediction and label vectors to enable easy performance measurement. An artifact of this

system is that the false negative and false positive numbers are equivalent, each signifying a binary misclassification. In total, we evaluate 396,166,430 ICD-9 classifications.

	Actual True	Actual False	
Predicted True	2,098,867	2,546,390	= 4,645,257
Predicted False	2,546,390	388,974,783	= 391,521,173
	= 4,645,257	= 391,521,173	n = 396,166,430

TABLE V

LSTM MODEL CONFUSION MATRIX

We see in table 6 some common performance metrics. The accuracy is very high, though in part due to a large class imbalance inherent in the dataset. On average there are only 7 ICD-9 codes attached to each patient out of 1070.

Metric	Value
Accuracy	0.98714
Precision	0.45183
Recall	0.45183
F1 Score	0.45183

TABLE VI

LSTM MODEL PERFORMANCE METRICS

A better measure of performance for this LSTM architecture is the ROC AUC plot. This is given in the next image. There, the predicted probabilities are used to graph the area under the curve. This has the advantage of using a soft rather than hard classification measure.

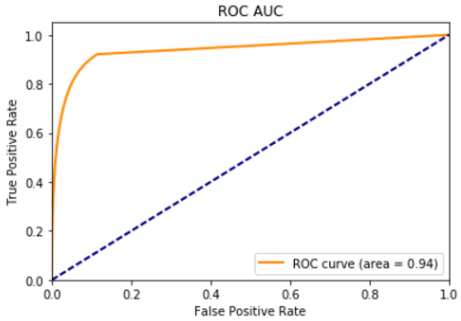

Fig. 7. LSTM ROC Results

VI. CONCLUSIONS

This project represents a prototype to a potentially viable product. With additional feature engineering using methods alternative to tf-idf and SVD and with enhancements in the deep learning architectures with augmentations such as attention mechanisms for LSTM models, the performance of the model could increase substantially. Similarly, resource constraints prevented the model from being trained on vocabulary extracted from all clinical notes and effected the number of model that could be experimented with.

We hope that this system can serve as a model for the future development of in-hospital systems that can provide doctors with the tools that empower them.

REFERENCES

[1] Weston J, Chopra S, Bordes A. Memory networks. arXiv preprint arXiv:1410.3916. 2014 Oct 15.

[2] Fries JA. Brundlefly at SemEval-2016 Task 12: Recurrent neural networks vs. joint inference for clinical temporal information extraction. arXiv preprint arXiv:1606.01433. 2016 Jun 4.

[3] Nigam P. Applying Deep Learning to ICD-9 Multi-label Classification from Medical Records.

[4] Miotto R, Li L, Kidd BA, Dudley JT. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. Scientific reports. 2016;6.

[5] Kim Y, Denton C, Hoang L, Rush AM. Structured Attention Networks. arXiv preprint arXiv:1702.00887. 2017 Feb 3.

[6] Perotte A, Pivovarov R, Natarajan K, Weiskopf N, Wood F, Elhadad N. Diagnosis code assignment: models and evaluation metrics. Journal of the American Medical Informatics Association. 2014 Mar 1;21(2):231-7.

[7] Subotin M, Davis AR. A system for predicting ICD-10-PCS codes from electronic health records. InProc BioNLP 2014 Jun 27 (pp. 59-67).

[8] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473. 2014 Sep 1.

[9] Choi E, Bahadori MT, Song L, Stewart WF, Sun J. GRAM: Graph-based Attention Model for Healthcare Representation Learning. arXiv preprint arXiv:1611.07012. 2016 Nov 21.

[10] Choi E, Bahadori MT, Sun J, Kulas J, Schuetz A, Stewart W. RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism. InAdvances in Neural Information Processing Systems 2016 (pp. 3504-3512).