Predicting Septicemia Using the MIMIC-III Database and Deep Learning Language Models

Group Members: Ivan Shu, Kelsey Luu, Taylor Shishido

Introduction

Electronic Health Records (EHR) provide a rich source of data that can be leveraged to draw actionable insights that benefit doctors, clinicians and patients. However, understanding EHR data has been a challenging task in natural language processing (NLP) due to the nature of unstructured data types and complex sources. Additionally, high quality clinical phenotyping from EHR data typically requires time-intensive expert review. In this project, we propose to implement an semi-supervised NLP model with the objectives of better understanding EHR data and deriving insights that can improve health outcomes without the need of a domain expert. The goal of this project is to use language models to predict the presence of a septicemia-related (aka sepsis-related) ICD9 code from the unstructured clinical notes of ICU patients. Specifically, we tackled the following two questions: 1) can we predict the number of septicemia-related ICD9 codes, if any, a patient has been assigned based on their clinical notes? This analysis is significant because we can potentially identify sepsis patient subtypes from their notes that are clinically significant, like subtypes stratified on survival time or length of stay in the hospital. Generally, we can perform deep phenotyping of sepsis.

Methods

Dataset

We created our datasets using two tables from the MIMIC-III database: 1) NOTESEVENTS, unstructured clinical notes of ICU patients, and 2) DIAGNOSES_ICD, ICD9 codes assigned to patients (Figure 2,1). Every entry in both tables is associated with both a hospital-specific patient ID (SUBJECT_ID) and hospital ID (HADM_ID), from which we created an unique ID. Although a patient could have clinical notes from different hospitals, we will not aggregate these together to describe the patient. We chose to do this because a patient might go to two different hospitals for very different reasons, such as for conception the first time and or septic shock the second time. We do not want their sepsis-unrelated notes for the first encounter to be assigned a positive binary outcome. Instead, we define two septicemia outcomes for every unique patient-hospital combination: a binary outcome of whether the patient-hospital ID has at least one septicemia-related code, and an ordinal outcome (multiclass outcome) of the number of ICD9 codes a patient was assigned. The binary outcome will be used in the downstream prediction task to answer the primary research question, and the ordinal outcome (multiclass outcome) will be used to answer one of the follow-up questions. Hereinafter, we use "patient" as a shorthand for patient-hospital combination.

There are 38 septicemia/sepsis-related codes in DIAGNOSES_ICD, which we found using a regular expression filtered for ICD9 code descriptions that contain “septic” or “sepsis” from the D_ICD_DIAGNOSES table, which is a key for all ICD9 codes in MIMIC. There were a total of 6,597 patients that had at least one and up to five sepsis codes, all of which were included in our dataset. To maintain class balancing, we randomly sampled 6,597 patients with no sepsis codes to include in our dataset. There are 15 categories of clinical notes, some of which are clinically irrelevant to sepsis diagnosis like Social Work and Nutrition. In order to downsample the volume, since training over two million notes would be too computationally intense, and to maintain high quality of the note content, we decided to utilize only the Radiology clinical notes (n=522,279 notes). These notes consist primarily of

fragmented sentences rather than numeric measurements and units and did not appear biasedly recorded for any patient subtype. Our final dataset consists of 13,194 patients and 164,230 radiology notes. We have a holdout set of 20% of the full dataset used for testing, and the remaining 80% was split, again 80-20, for model training and validation. To prevent models from merely learning patient-specific rather disease-relevant information, every patient has all of their clinical notes in one of the three datasets (i.e. patients do not have notes in both the testing and training set).

Figure 1. DIAGNOSES_ICD.csv example. Columns used: SUBJECT_ID, HADM_ID, ICD9_CODE

ROW_ID <int>	SUBJECT_ID <int>	HADM_ID <int>	SEQ_NUM <int>	ICD9_CODE <chr>
1297	109	172335	1	40301
1298	109	172335	2	486
1299	109	172335	3	58281
1300	109	172335	4	5855
1301	109	172335	5	4254
1302	109	172335	6	2762

Figure 2. NOTESEVENTS.csv, part 1. Columns used: SUBJECT_ID, HADM_ID, CATEGORY, ISERROR, TEXT

Description: data.table[,11] [6 × 11]

ROW_ID <chr>	SUBJECT_ID <dbl>	HADM_ID <dbl>	CHARTDATE <S3: IDate>	CHARTTIME <S3: POSIXct>	STORETIME <S3: POSIXct>	CATEGORY <chr>	DESCRIPTION <chr>	CGID <int>	ISERROR <int>
174	22532	167853	2151-08-04	<NA>	<NA>	Discharge summary	Report	NA	NA
175	13702	107527	2118-06-14	<NA>	<NA>	Discharge summary	Report	NA	NA
176	13702	167118	2119-05-25	<NA>	<NA>	Discharge summary	Report	NA	NA
177	13702	196489	2124-08-18	<NA>	<NA>	Discharge summary	Report	NA	NA
178	26880	135453	2162-03-25	<NA>	<NA>	Discharge summary	Report	NA	NA
179	53181	170490	2172-03-08	<NA>	<NA>	Discharge summary	Report	NA	NA

TEXT <chr>
Admission Date: [**2151-7-16**] Discharge Date: [**2151-8-4**]\n\n\nService:\nADDENDUM:\n\nRADIOLOGIC STUDIES: Radiologic studies also included a chest\nCT, which...
Admission Date: [**2118-6-2**] Discharge Date: [**2118-6-14**]\n\nDate of Birth: Sex: F\n\nService: MICU and then to [**Doctor Last Name **] Medicine\n\nHISTORY OF PR...
Admission Date: [**2119-5-4**] Discharge Date: [**2119-5-25**]\n\n\nService: CARDIOTHORACIC\n\nAllergies:\nAmlodipine\n\nAttending:[**Last Name (NamePattern1) 15...
Admission Date: [**2124-7-21**] Discharge Date: [**2124-8-18**]\n\n\nService: MEDICINE\n\nAllergies:\nAmlodipine\n\nAttending:[**First Name3 (LF) 898**]\nChief Com...
Admission Date: [**2162-3-3**] Discharge Date: [**2162-3-25**]\n\nDate of Birth: [**2080-1-4**] Sex: M\n\nService: MEDICINE\n\nAllergies:\nPatient recorded as having N...
Admission Date: [**2172-3-5**] Discharge Date: [**2172-3-8**]\n\nDate of Birth: [**2109-10-8**] Sex: F\n\nService: NEUROSURGERY\n\nAllergies:\nNo Known Allergies / A...

Text Preprocessing

To address the messiness of clinical notes, we removed bracketed information, numbers, punctuation, stop words, carriage returns, and newline characters. We then lemmatized words and tokenized notes by word using a dictionary of the top 10,000 words. Most of the preprocessing was performed using the Natural Language Toolkit (https://www.nltk.org/) and tokenization done by a tensorflow.keras Tokenizer object. We used the tokenizer to convert tokens to integers and padded them to length 512.

Models and Results

Baseline models: Bag of Words and Word2Vec

We constructed two baseline models for the task that, unlike RNNs, do not encode long-range sequential information: bag of words (BOW) and Word2Vec. The BOW approach, specifically latent semantic analysis, counts the frequency of unique terms in each note and generates similar embeddings for notes that have similar unique terms. We applied a term frequency inverse document frequency (TFIDF) transformation on the entire training set of notes and applied matrix factorization to obtain 500-dimension embeddings for each of the 10,000 words. Word2Vec learns word embeddings by observing the nearby neighbors of words in each note and tries to generate similar embeddings for words that have similar neighbors. Using the first 50,000 clinical notes from the training set (~47%) due to Google Colab memory

constraints and looking at the two nearest nearby on each side of a word (four neighbor words total), we generated 100-dimension embeddings of each word. For both embedding methods, we chose four sepsis-specific and two nonspecific terms and looked at the top 10 words with the shortest Euclidean distance in the embedding space to each target word (Figure 4). The control words, which are not specific to septicemia and likely occur in sparser contexts, are similar to only other generic medical terms: "dressing", "hrs", "amount" and etc. Although we still see this pattern with similar words to the sepsis-specific terms, we also see that similar words include causes of sepsis like pneumonia, hydronephrosis (kidney swelling from blocked urine), hepatoma (liver cell cancer), and complications like gangrene chest. This suggests that our baseline models are somewhat recognizing and learning terms related to sepsis or septicemia.

SEPSIS		**SEPTICEMIA**		**SHOCK**	
anterosuperior	dic	comply	accidentally	moderatetolarge	cardiogenic
pnumonia	pneumococcal	cost	tuesday	mvr	septic
abundant	mellituscoagulopathy	cea	yearold	unaltered	hypovolemic
abnormalaties	noninsulin	carcinomatosis	extract	home	mellitusseptic
gre	hepatico	motion	think	painchest	deficency
serial	tapped	infarctiontelemetry	cbd	subtraction	cholangitisseptic
hydronephrosis	hepatoma	obtai	extraction	maroon	cs
deviation	sepsishypotension	absorption	age	orally	hypotensionrenal
arteriography	sepsisacute	code	perc	pseudo	bradycardiahypotensionsepsis
BRAIN (control word)		**CHEST (control word)**		**BLOOD**	
ante	diffusionweighted	bihilar	portable	fixator	bright
amount	anoxic	diseasemultiple	ap	diskitis	red
impinge	stereotaxis	tx	reason	administrate	pool
cental	death	dressing	condition	bleeds	autologous
additional	brainstem	anastamotic	final	coul	label
loosening	susceptibility	de	medical	gangrenechest	brbpr
far	flair	obstructionileus	examination	composite	product
nonionic	timeof	btca	number	consis	rbc
atraumatic	dwi	hrs	report	ambient	maroon

Figure 3. Six target words (bold) and top similar words in descending order in the Word2Vec (yellow) and LSA (green) embedding spaces. *Control words are conceptually unspecific to sepsis.*

We computed note embeddings by averaging all the word embeddings in each note. These embeddings were passed into a simple NN architecture of one dense and one ReLu layer and a final output layer (sigmoid, softmax, or linear) that matched based on the downstream prediction (binary, multiclass, or regression, respectively). The binary and multiclass results for the baseline models are reported in the model summary. Surprisingly, the LSA embeddings consistently performed better for the downstream tasks compared to Word2Vec. Although Word2Vec embeddings should theoretically be more successful, their worse performance might be caused by their smaller dimension size, which captures limited information compared to a larger dimension, or that they were learned from only a subset of the training data. Compared to the classifier model accuracies (75-80%), the regression task performed poorly.

Table 1. Regression performance metrics of the baselines models on the validation set.

	Mean squared error	**Mean absolute error**	**Correct predictions**
Word2Vec	**0.72137**	**0.66679**	**51%**
LSA	**0.472789**	**0.56797**	**59.7%**

Basic Neural Networks

Stepping up in complexity, we constructed two basic neural network architectures: LSTM and CNN. We chose an LSTM framework due to the sequential nature of the data, and CNN because of its ability to learn translationally invariant patterns and extract high level features. These models take input as a matrix where each note is represented by a 512-element integer vector. For binary classification, the output layer for each of these models consisted of a single sigmoid-activated neuron, and for multiclass classification, the output was a softmax-activated layer with 1 neuron per class (n=6). The two architecture summaries are as follows:

1. LSTM: Input layer, embedding layer with 100-dimensional output, 128-unit LSTM layer, 256-unit dense layer, dropout layer with dropout rate of 0.5, and output layer.
2. CNN: Input layer, embedding layer with 100-dimensional output, 1D-Convolution layer with 128 filters and relu activation, dropout layer with dropout rate of 0.8, 1D global max pooling layer, and output layer.

Both models were compiled and trained using either the binary or categorical (multiclass) cross entropy loss functions, and Adam optimizer in Keras. The models were trained for 5 epochs with a batch size of 128 and evaluated on both the validation set (n=26,056) as well as the test set (n=32,539). Overall, both the LSTM and CNN performed well considering the training time and model complexity, yielding slightly lower evaluation metrics than the BERT models with only a fraction of the parameters (Table 5,6).

To guide model interpretation, we projected the output of an intermediate layer of the LSTM model into a lower dimensional space and used a UMAP plot to visualize where these clinical notes lie relative to each other according to their learned embeddings (Figure 4). We see that the radiology notes originating from patients with at least 1 sepsis ICD code are highly separable from those that originate from non-sepsis patients. This suggests that the LSTM has learned a relevant embedding for our classification task.

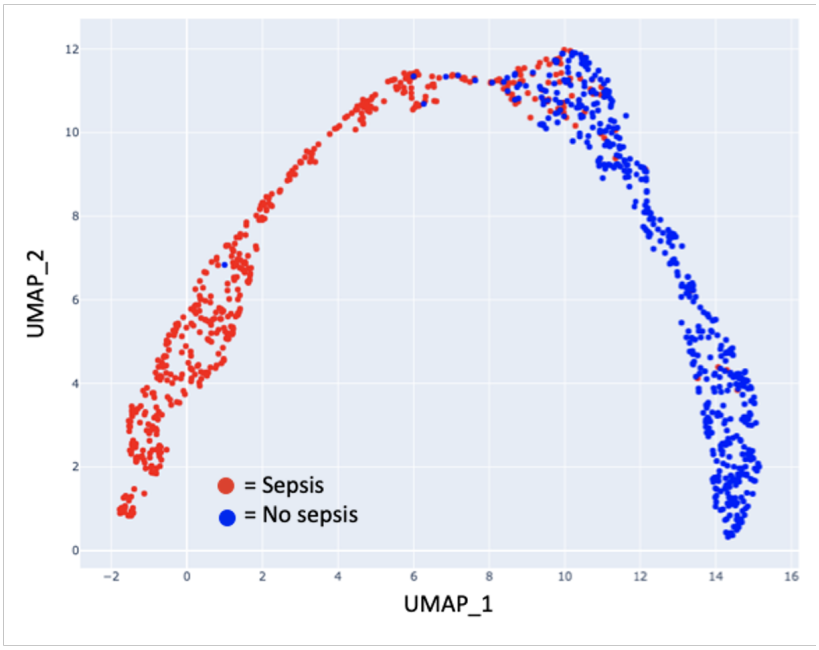

Figure 4. UMAP visualization of the LSTM layer output for 1000 training notes. *Sepsis notes are colored red and non-sepsis notes are colored blue.*

To assess whether the model is simply stratifying notes on the presence or absence of the word "sepsis", the top 100 test set clinical notes with the highest predicted probability of originating from a sepsis patient were extracted for inspection. Of these 100 notes, 34 explicitly contained the word "sepsis", meaning it may be necessary to filter out any sepsis-related words from the notes prior to training for more meaningful predictions in the future, though it does not appear that the model is exclusively reliant on the mention of sepsis for classification.

DistilBERT

To leverage the state-of-the-art pre-trained language models, we decided to use BERT, a transformer model architecture from HuggingFace. Specifically, DistilBERT was implemented with some fine-tunings of our MIMIC-III datasets and downstream task of predicting the presence of Septicemia-related ICD9 codes. The main reason for choosing DistilBERT was because it is light-weight, energy-efficient and has shown great and comparable performance in contrast to BERT. A different tokenization process unlike RNN models was performed due to the specifics of the BERT model implementation. During training, we chose 3 epoch iterations with the expectation that only minimum fine-tuning is required. The model was evaluated on our binary outcome classification and multiclass classification tasks (Table 2, 3).

Table 2. DistilBERT performance on presence of septicemia ICD9 code (binary-outcome classification)

DistilBert	Accuracy	Precision	Recall	F1	AUC	Time (min)	# param
Train set	0.96128	0.97258	0.94969	0.96100	0.95446	81.10	66,955,010
Val set	0.84363	0.87501	0.80507	0.83858	0.81807	81.10	66,955,010
Test set	0.83107	0.86399	0.77834	0.81893	0.83071	81.10	66,955,010

Table 3. DistilBERT performance on occurrences of septicemia ICD9 code (multiclass-outcome classification)

DistilBert	Accuracy	Precision	Recall	F1	Time (min)	# param
Train set	0.49770	0.49760	0.48734	0.48746	97.74	66,955,010
Val set	0.49548	0.49335	0.48992	0.49162	97.74	66,955,010
Test set	0.50917	0.50025	0.50446	0.50234	97.74	66,955,010

Bio_ClinicalBERT

Although transfer learning has shown great successes compared to models built from scratch, the generalizability to other datasets or prediction tasks depends heavily on the context the model was pre-trained on. By looking at DistilBERT performance, the performance on multiclass outcome with occurrences of septicemia ICD9 was quite poor. This is likely due to the fact that DistilBERT was only pre-trained on English Wikipedia and *Toronto Book Corpus*, divergent from clinical background in general. We then decided to look at other more context-relevant BERT models. One article published by Alsentzer et al., showed that they had trained the BERT model specifically with the MIMIC-III dataset called Bio_ClinicalBERT[1]. This model was built upon BioBERT and further fine-tuned with MIMIC-III notes and discharge summaries. To implement Bio_ClinicalBERT with our prediction tasks, we froze the weights for all the layers except for the output layer with additional fine-tunings. Similarly, Bio_ClincalBERT was only trained with 3 epoch iterations.

Table 4. Bio_ClinicalBERT performance on presence of septicemia ICD9 code (binary-outcome classification)

Bio_ClinicalBERT	Accuracy	Precision	Recall	F1	AUC	Time (min)	# param
Train set	0.96747	0.98187	0.95281	0.96712	0.96753	85.04	108,310,272
Val set	0.84303	0.86251	0.81872	0.84004	0.84319	85.04	108,310,272
Test set	0.82633	0.85140	0.78288	0.81538	0.82548	85.04	108,310,272

Table 5. Bio_ClinicalBERT performance on occurrences of septicemia ICD9 code (multiclass-outcome classification)

Bio_ClinicalBERT	Accuracy	Precision	Recall	F1	Time (min)	# param
Train set	0.97369	0.99064	0.95664	0.97334	81.0	108,310,272
Val set	0.84333	0.87940	0.79829	0.83688	81.0	108,310,272
Test set	0.82279	0.86264	0.75946	0.80776	81.0	108,310,272

Clearly, Bio_ClinicalBERT has improved its multi-outcome classification performance significantly compared to DistilBERT with similar binary-outcome classification performance. To further visualize how Bio_ClinicalBERT models interpret input sentences, we next visualized the attention heads from an output layer with this input sentence “he has experienced acute chronic diastolic heart failure in the setting of volume overload due to his sepsis.”

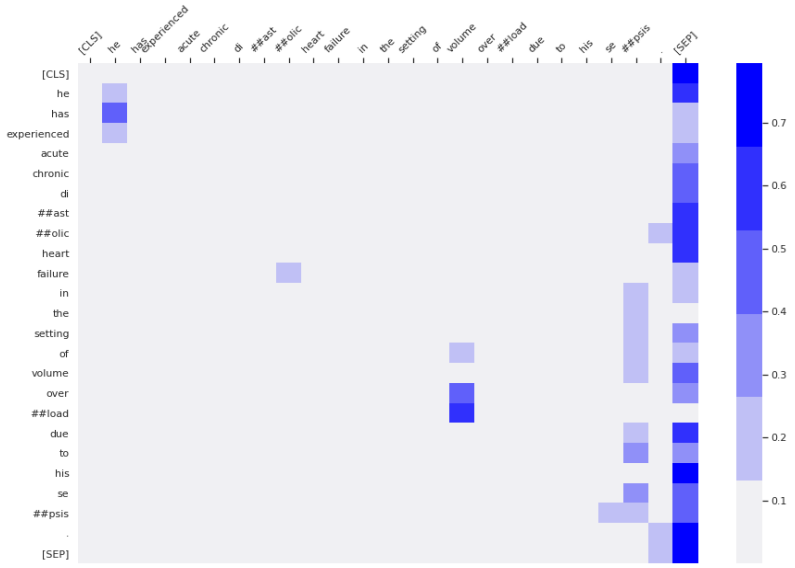

Figure 5. Bio_ClinicalBERT model attention heads visualization

Compared to other tokens, "##psis" token takes into account the context it is in by placing many non-negligible attention weights on other tokens. This phenomenon might have an association with downstream prediction tasks. The output layer, however, is usually deep, so the interpretation becomes abstract and this visualization cannot be taken too strictly rather than a direct guideline.

Model Summary

We then compared all models implemented in this project by training time, model complexity (number of parameters) and performance on the test set.

Figure 6. Models training time and complexity comparison

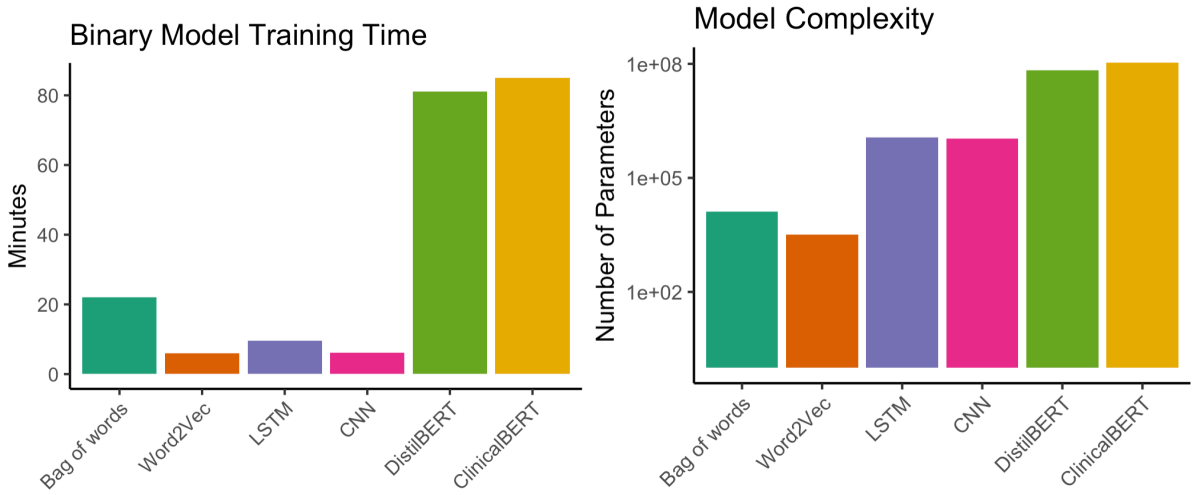

Table 6. All models performance on test set with binary-outcome classification

Models (Binary Outcome)	Accuracy	Precision	Recall	F1	AUC

Bag of words	0.800	0.825	0.751	0.786	0.881
Word2Vec	0.752	0.749	0.742	0.746	0.883
LSTM	0.809	0.898	0.689	0.780	0.893
CNN	0.696	0.640	0.872	0.738	0.809
DistilBERT	0.831	0.864	0.778	0.819	0.831
Bio_ClinicalBERT	0.826	0.851	0.782	0.815	0.825

Table 7. All models performance on test set with multi-outcome classification

Models (Multi Outcome)	Accuracy	Precision	Recall	F1
Bag of words	0.629	0.736	0.480	0.581
Word2Vec	0.569	0.707	0.448	0.548
LSTM	0.659	0.745	0.529	0.619
CNN	0.723	0.813	0.588	0.683
DistilBERT	0.509	0.500	0.504	0.502
Bio_ClinicalBERT	0.823	0.863	0.759	0.808

Evidently, BERT models are the most complex models, followed by LSTM and CNN, and admittedly they also take significantly longer time to train. Bag of words and Word2Vec spend a similar amount of time during training. As for performance on binary-outcome classification, they all performed equally with BERT models being the best, followed by LSTM and Word2Vec models. As for multi-outcome classification, Bio_ClinicalBERT model outperformed all other models and LSTM was the second. Although Bio_ClinicalBERT shows absolute dominance in both classification tasks, one caveat is the time and energy resources required during the model training process. Models need to be carefully chosen for specific clinical needs. For example, if accuracy is the only metric, clinical researchers might need to choose the best-performance model at the expense of resources.

Additional Analysis

Initially, there was concern that our models were only learning to recognize keywords like “sepsis” and “septicemia”, and primarily relying on these to make correct predictions. To explore this, we used the test dataset evaluated on Bio_ClinicalBERT. From the 32,539 notes, only 3,288 notes (~10%) contained the words 'sepsis' or 'septicemia'. From the 26,990 notes that were correctly predicted, 3,008 notes (~11%) contained the two key words. These observations suggest that because so few of the radiology notes contain phenotype keywords, the language models are learning to predict sepsis beyond obvious word cues, which is the level at which a layperson or potentially physician might perform as a classifier.

We also explored further whether the portion of patients that were false negatives, meaning they had a sepsis ICD but were predicted to not have one, was significantly different compared to the whole test population. To determine whether specific sepsis ICD codes were being incorrectly predicted more than others, we took the Bio_ClinicalBERT predictions on the test dataset, extracted all of the patients corresponding to the test set, and compared the breakdown of sepsis ICD codes between the whole test population (n=2,506 patients) and the false negatives (n=531). The distribution of sepsis codes for the false negatives is very similar to the whole population (Table 8). This suggests that none of the sepsis ICDs are associated with lower or higher rates of incorrect prediction than expected.

Table 8. Population breakdown of different sepsis codes for all test patients and false negative patients.

ICD Code Description		% of all patient sepsis codes	% of all FN patient sepsis codes
Severe sepsis		0.27	0.25
Unspecified septicemia		0.25	0.27
Septic shock		0.17	0.14
Sepsis		0.09	0.07
Methicillin susceptible Staphylococcus aureus septicemia		0.04	0.05
Septicemia due to escherichia coli [E. coli]		0.03	0.03
Other septicemia due to gram-negative organisms		0.03	0.04
Streptococcal septicemia		0.03	0.03
Septicemia [sepsis] of newborn		0.02	0.04
Other specified septicemias		0.02	0.02
Other staphylococcal septicemia		0.01	0.01
Methicillin resistant Staphylococcus aureus septicemia		0.01	0
Septicemia due to pseudomonas		0.01	0.01
Pneumococcal septicemia [Streptococcus pneumoniae septicemia]		0.01	0.01
Septicemia due to anaerobes		0.01	0.01
Septic pulmonary embolism		0	0
Staphylococcal septicemia, unspecified		0	0.01
Septic arterial embolism		0	0
Septicemia due to gram-negative organism, unspecified		0	0
Septicemia due to serratia		0	0
Postoperative shock, septic		0	0
Septicemia due to hemophilus influenzae [H. influenzae]		0	0

To determine whether patients with a certain number of ICD codes were being incorrectly predicted more than others, we compared the breakdown of number of ICD codes per patient between the whole test population and the false negatives (Table 9). The majority (defined as >75%) of all test patients have two or three sepsis ICDs. However, the majority of the false negative patients have one or two ICDs. This suggests that patients with fewer sepsis ICDs tend to be mislabeled by the classifier at higher rates. This is not surprising, as patients with more ICDs might have had more severe cases of sepsis that were described more thoroughly in their notes or that resulted in a higher volume of clinical notes.

Table 9. Population breakdown of number of sepsis ICDs for all test patients and false negative patients.

# of sepsis ICDs	% of all patients	% of FN patients
1	0.2	0.36
2	0.44	0.41
3	0.34	0.23
4	0.02	0.01
5	0	0

Discussion

There are some challenges we have faced when conducting this project. The first is that in the dataset, there are around 200,000 patients that are missing either subject ID or specific hospital-stay ID. We had to drop those samples and might have run into the risk of losing important samples. Although this might pose some issues, our data size is still considered as large-scale. In particular, because clinical notes are very unstructured, each note is lengthy on average and can vary dramatically from patient to patient. This caused us to have hardware-specific difficulties, such as GPU time-out when loading the data into the Colab environment.

A limitation of this model is that it was trained on retrospective data. We did not have the ability to test our models real-time on patients currently in the hospital, who had or had not yet received a billing code for sepsis. Additionally, there could be issues involving external biases, like only collecting data from academic medical centers or hospitals only, as well as variable data quality that may reduce the robustness of our model. Another limitation is that the data that this model was trained on did not provide temporal information for diagnoses, so we were unable to exclusively train on clinical notes for a patient that were written after a formal sepsis-related diagnosis. Instead, we operate under the assumption that all clinical notes for a sepsis patient (before or after diagnosis) are informative for our prediction task, which may or may not be true in practice. These factors may also impact the generalizability of our model to ICU data collected by institutions outside of those in the MIMIC-III training dataset. Further evaluation should be done to assess model performance on additional data sources.

Regarding model interpretation, another potential avenue to explore is sentiment analysis. We can extract the original text of clinical notes from similar embeddings generated by our language models and manually compare the content of the clinical notes to identify any semantic or syntactic similarities. By clustering notes in embedding space, we may also be able to identify sepsis subtypes that were not otherwise specified or explicitly diagnosed. Additionally, we can perform saliency mapping on high-scoring clinical notes to determine which input tokens within these notes contribute the most to changes in the output sepsis probability. Lastly, it may be informative to predict the presence of individual sepsis-related ICD codes as opposed to aggregating 38 codes into a broad “sepsis” label, as some codes may be more difficult to predict than others.

Citations:

1. An introduction to latent semantic analysis, Quantitative Approaches to Semantic Knowledge, 2009, Landauer et al.
2. Efficient Estimation of Word Representations in Vector Space, Computation and Language, 2013, Mikolov et al.
3. MIMIC-III: a freely accessible critical care database, Nature, 2016, Johnson et al.
4. Publicly Available Clinical Embeddings, arXiv, 2019, Alsentzer et al.

Contributions

Overall, the work was distributed equally for the project. All group members contributed equally to writing the proposal, project check-in, project presentation, and final report. Everyone helped to brainstorm the research questions and dataset. Taylor did the preprocessing of the clinical notes data and built the bag of words and Word2Vec models, Kelsey built the CNN and LSTM models, and Ivan built the DistilBERT and ClinicalBERT models. We each did further analysis of the models that we built. The discussion of the report was brainstormed by all of us and written up by Kelsey.