Reproducibility Project Instructions for CS598 DL4H in Spring 2023

Sandeep Yerramsetty and Deep Shah
{sy40, deepss2}@illinois.edu

Group ID: 104
Paper ID: 163
Presentation link: https://youtu.be/Cod4AFHHfhQ
Code link: https://github.com/deepss2/CS598_DLH_Project

1 Introduction

We plan to reproduce the paper "Comparing deep learning and concept extraction based methods for patient phenotyping from the clinical narrative" (1) as part of CS598 Deep learning healthcare coursework, Fall'23 UIUC.

Recent advancements in healthcare and the adoption of Electronic health records (EHR) have enabled researchers to do in-depth analysis. EHR data consist of structured data such as patient demographics, laboratory tests, medication, etc, and unstructured data such as clinical notes. Though Structured data are easier to analyze and don't require much pre-processing as compared to unstructured data, unstructured data does have much richer information and can be very useful if used appropriately.

Patient phenotyping is one such classification task that classifies whether the patient has a particular condition or is likely to develop one in the future. Patient phenotyping is one of the crucial tasks involved at various stages and can help reduce the amount of time spent by clinicians daily doing it.

2 Scope of reproducibility

Claim 1: CNN has been shown to better detect patient phenotype from the clinical notes compared to other baseline models in our case n-gram and Bag-of-word (BOW) models.

Here's how we will test this hypothesis:

- We will be evaluating AUC score to compare the performance across the models.
- Given there are multiple phenotypes, we will compare the max score across various CNN models vs the N-gram model.

Claim 2: CNN models add value by having a convolution of width greater than one, while n-gram models don't show the same. Plan to test the reproducibility claim:

- Comparison of AUC score comparison between CNN model with width one vs width 1-2.

3 Methodology

3.1 Model descriptions

The model is the CNN model on a text which uses the same foundation as applying a convolution neural network on images. Filters are learned that merge patches of a pixel into a single value that represents some meaning or concept. Here, CNN on the text will take a sub-string of words and learn the concepts which will be used to predict the phenotype in our case.

An overview of the model is shown in Fig 1.

The input to the CNN model is a sequence of words projected on pre-trained embedding. Pre-trained embeddings are used since it has proven to provide a better improvement on various NLP tasks since it helps model generalize better.(2)

CNN of different widths is applied on top of this to learn the concepts from the model. Latent concepts are learned from the output of CNN, since the concept could be present anywhere in the discharge summary text we will apply MaxPooling on top of the available feature map to compute the max signal from the output of CNN. Dropout is applied on the top of that to ensure that the model doesn't overfit and generalizes well.

The output of MaxPooling is passed to a fully connected network which predicts the score of whether the phenotype exists in the model or not.

One important part of the model architecture is how the model training is done for all the phenotypes. A separate model is trained per phenotype so that models don't interfere with each other and optimal performance is achieved.

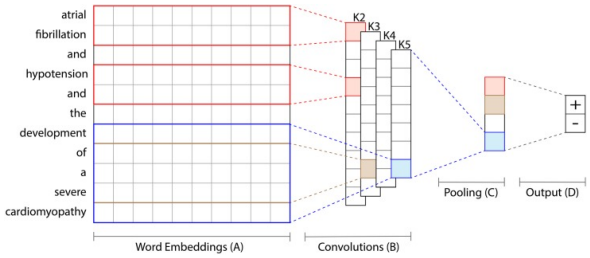

Figure 1: CNN architecture with Pooling

3.1.1 Word Embedding parameters

Embedding for the words is learned using all the discharge summaries available. Word2Vec model is used to learn the embeddings. Here are the hyper-parameter we used for learning the word2vec model:

- Window size = 10
- Negative examples = 10
- Num of iterations = 15
- Drop words occurring $<$ 5 times

Pre-processing of text was done to remove punctuation and add space where required.

3.2 Data descriptions

We will be using two datasets for training the model and embeddings.

1. MIMIC-III data: It contains discharge summaries of various patients keyed by patient id, hospital id, and admission time. Patients can have multiple summaries since the person could be admitted multiple times. It also contains data regardings reports, etc but we only care about the discharge summaries and filter others out. We downloaded the dataset from PhysioNet, but due to privacy reasons, we won't be uploading this data into our GitHub repo.

2. Annotation data: The author of the paper generated high-quality labels using clinical researchers, junior residents, senior residents, and practicing intensive care medicine physicians. The author has provided the labeled 1611 labeled data on the open-source repository. Due to privacy reasons discharge summaries aren't available in the data and we need to join it with MIMIC-III to get discharge summaries. There could be multiple entries for <patientId, hospitalID> since a patient could have been admitted more than once.

Our plan was to join both datasets using the unique key <patientID, hospitalID, chartTime>but chartTime wasn't properly populated in the annotation data provided by the author.

This will pose a challenge on how to join the data since the number of entries for a patient could be different in MIMIC-III and Annotation data. To overcome that, we will merge all the summaries of the patient and use that as a summary of the patient. This will reduce the number of annotated example counts we have since multiple entries for a patient will be consolidated into single entry post this. Describe the datasets you used (with some statistics) and how you obtained them.

Table 1: MIMIC-III stats

Number of discharge summary	59652
Number of distinct patient	41127

Table 2: Annotated data stats

Number of discharge summary	1610
Number of distinct patient	1045
Number of distinct patient and hospital	1550

3.3 Hyperparameters

To be able to reproduce the experiment to a greater extent, we tried to use the same parameter. Here are some of the hyper-parameters:

- Batch size: 32
- Adadelta optimizer with rho: 0.95
- Dropout rate: 0.5
- Embedding width: 100
- Number of feature map learned per convolution width: 100
- Initialized weight uniformly between +/-0.05.
- Words not present in word2vec were initialized randomly between +/-0.25.

3.4 Implementation

The original code by the author has been published here(3). Unfortunately, even though the code is available on GitHub we won't be able to use it due to multiple reasons:

1. Author didn't publish their code to compute the embeddings from the MIMIC-III discharge summaries, they published the drive link to the summaries but it is not working last we checked.

2. Author has used Lua and the code has been written in 2017, many developments have been made since then, and hence we will implement the logic using Pytorch and Python packages.

Our code can be found here on GitHub.

3.5 Computational requirements

The author didn't mention training requirements in the original paper. Since there are only 1600 annotations, we should be able to train the model on a local computer. We will be using torch and will train our model on MacBook Pro. Specs of the computer are:

- RAM: 32GB
- Processor: 2.2 GHz 6-Core Intel Core i7
- GPU: Radeon 4GB
- Average epoch time: 37 seconds
- Time per phenotype: 12m 20 sec
- Training epoch: 20

4 Results

4.1 Evaluation Metrics

The original paper reported Precision, Recall, F1, and AUC scores. One of the problems with reporting precision and recall is there could be multiple different pairs achieving the same F1 score and hence we will use AUC or F1 score to compare the models.

4.2 Reproducting Result

4.2.1 N-gram model

N-gram-based models were trained using Logistic regression to generate the prediction for the model. Different models with n-grams ranging from 1 to 4 were trained and the best model was picked having the highest F1 score. Results can be found here at Tab3.

Table 3: N-gram F1 score of reproduced vs original

Phenotypes	Reproduced	Original
Advanced.Cancer	91	90
Advanced.Heart.Disease	87	86
Advanced.Lung.Disease	90	79
Alcohol.Abuse	85	89
Chronic.Neuro	70	72
Chronic.Pain	71	68
Depression	79	78
Obesity	75	72
Other.Substance.Abuse	83	90
Psychiatric.Disorders	75	77

The AUC score is quite close between reproduced model and the score reported by the authors for most of the phenotypes.

Based on the above, we can say that we are able to reproduce the baseline n-gram models.

4.3 CNN model

Here we use the CNN architecture as described above. Since most of the gain is coming when going from the CNN filter of width 1 to width 1-2, we will focus on only two widths 1 and 1-2.

We compare the AUC reported by the paper vs what we were able to reproduce in Tab 4 and 5.

Table 4: AUC score of reproduce vs original of CNN model width 1

Phenotypes	Reproduced	Original
Advanced.Cancer	93	85
Advanced.Heart.Disease	90	86
Advanced.Lung.Disease	80	83
Alcohol.Abuse	90	80
Chronic.Neuro	76	74
Chronic.Pain	72	72
Depression	93	71
Obesity	74	69
Other.Substance.Abuse	87	87
Psychiatric.Disorders	90	77

- AUC score is same for most of the phenotypes for both the CNN model we trained with allowed width 1 (Tab 4) and 1-2 (Tab 5).

Table 5: AUC score of reproduce vs original of CNN model width 1-2

Phenotypes	Reproduced	Original
Advanced.Cancer	96	94
Advanced.Heart.Disease	93	90
Advanced.Lung.Disease	86	89
Alcohol.Abuse	93	89
Chronic.Neuro	79	80
Chronic.Pain	76	78
Depression	95	92
Obesity	84	100
Other.Substance.Abuse	88	97
Psychiatric.Disorders	89	95

- The largest mismatch is for phenotype Obesity for CNN width 1-2, and phenotype Depression for CNN width one.
- We can say that the experiment is able to reproduce the result as expected since it is matching for most of the phenotypes.

There could be various reasons justifying the differences between retrain and reproduced:

- Difference in the train vs test data meaning there could be differences due to data variance given there are only 1.6k labeled examples. i.e. different test examples used by the author vs us.
- There could be multiple summaries for a patient and due to the incompleteness in the labeled data provided by the author (explained in the Data section) we couldn't identify which summaries were used. We approximated this by concatenating all the summaries for a patient and using that as input.

4.3.1 Analyze Claims by the paper

There were two claims by the paper

1. CNN has been shown to better detect patient phenotype from the clinical notes compared to other baseline models in our case n-gram and Bag-of-word (BOW) models.

   From the 2, we can see that the AUC score of the reproduced CNN always outperforms the reproduced N-gram model. We already observed that we are able to reproduce the experiment up to a greater extent and hence can say that the claim is verified by our experiment.

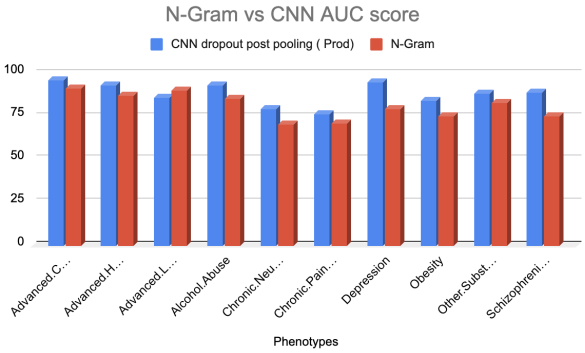

Figure 2: N-Gram vs CNN AUC Score

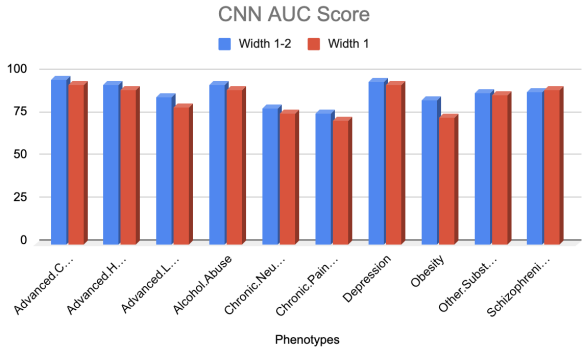

Figure 3: CNN AUC Score

2. CNN models add value by having a convolution of width greater than one, while n-gram models don't show the same.

   From the 3, the AUC score of the CNN with width 1-2 always outperforms the one with width 1, which supports the claim.

4.3.2 Ablation Study

Discharge summaries are collections of words and recent advancements have shown NLP-based Deep learning techniques to excel at such tasks. We decided to use the LSTM network since the LSTM network has the ability to remember and carry forward important information from the past and the important terms/concepts could occur anywhere in the text.

We performed two experiments using the LSTM network to understand the author's proposed architecture better.

AUC score for all the ablation studies can be found in Tab 6

1. **LSTM** We pass the output embedding of the last token to the Linear layer followed by Sigmoid. Output embedding of the last token is expected to learn the significant concept from

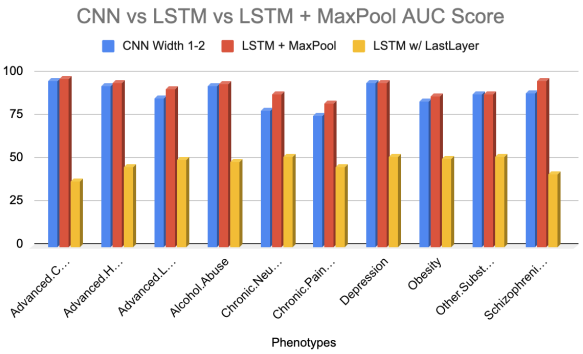

Figure 4: CNN vs LSTM AUC Score

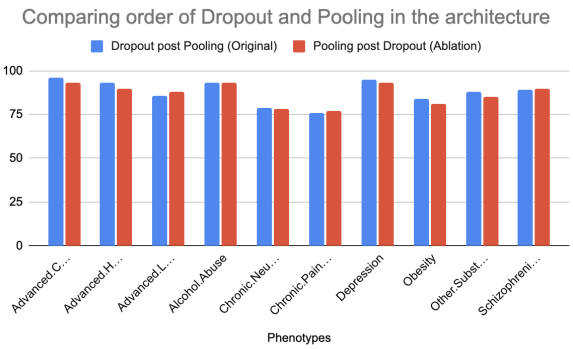

Figure 5: MaxPool vs Dropout order AUC Score

the whole discharge summary and help predict the phenotype.

- AUC score of the LSTM is quite low compared to the CNN reproduced model as seen in Fig 4.
- That means it is difficult for the model to learn the occurrence of the important concepts in the sentence and carry that forward to the end using LSTM. This suggests that MaxPool is important in the architecture of our use case. We can make a general statement since we don't know whether LSTM will be good enough if there would have been large data set.

2. **LSTM + MaxPool** We apply max-over-time-pooling over the output embedding of all the tokens, which is then passed to the Linear layer followed by Sigmoid.

   A concept can be in multiple ways and with different numbers of words. CNN limits us since we need kernel filters of different widths to capture various lengths, LSTM on the other hand takes the concept till now and augments it with the current words. This saves us from tuning the cnn filter widths and generalizes better.

   - AUC score of the LSTM + MaxPool is at least as good as with CNN width 1-2 while outperforming at some phenotypes (Fig 4).
   - This suggests that the main component of the architecture is MaxPool, while CNN can be easily replaced with LSTM-like architecture. This can save us from having different hyper-parameter for cnn kernel widths.

3. **CNN dropout followed by pooling** Author has done Pooling followed by Dropout.

   Mathematically, during inference, both of them are the same and hence we do not expect any performance difference by swapping the sequence of them.

   - AUC score performance is same for both the architecture i.e. dropout followed by pooling vs pooling followed by dropout as seen in fig 5

Table 6: AUC score of ablation

Phenotypes	LSTM	LSTM + MaxPool	CNN Dropout followed by Pooling
Advanced Cancer	38	97	93
Advanced Heart Disease	46	95	90
Advanced Lung Disease	50	91	88
Alcohol Abuse	49	94	93
Chronic Neuro	52	88	78
Chronic Pain	46	83	77
Depression	52	95	93
Obesity	51	87	81
Other Substance Abuse	52	88	85
Psychiatric Disorders	42	96	90

4.4 Word2Vec

Word2Vec embeddings for the token have been trained using all the discharge summaries available in MIMIMC-III.

From the table, we can see that the embeddings learned by word2vec are representing similar words closer to each other.

Here, we looked at the top 5 nearest words for different medical words.

Table 7: Word2Vec Top 5 similar words

Surgery	operation, surgeons, procedures, revascularization, surgery
heart	sedimentation, sed, multiorgan, reanl
drug	ethanol, alcohol, drugs, tobacco, substance

5 Discussion

The key takeaways from the experiments and ablation study are:

- The architecture proposed by the author is able to perform much better than the N-gram model which is the baseline.
- Though the CNN model is able to learn the feature across text, the most critical component is the MaxPool since we saw that only LSTM w/o MaxPool wasn't performing well.
- Upon quickly searching about what should be the order of the MaxPool and Dropout layer, there were many articles saying one or the other. Our ablation study showed us that the ordering doesn't matter for our data and either works fine.

Some of the difficulties we faced during the study were:

- Author didn't provide the chart time of the discharge summary and hence we couldn't generate the exact training data. We circumvent this by contacting all the summaries of the patient and treating that as one data point.
- The task at hand wasn't end-to-end, i.e. the whole training happened in one go. It was two-step which involved generating the embedding of the words and learning the model post that. One difficulty which arises due to that was when the model wasn't performing it was hard to pinpoint whether it was due to embedding not being good enough or an issue with model training. One thing we did to alleviate this was to try out various health-related words on our word2vec-trained model to verify it is working as expected.
- Understanding why the precision and recall weren't matching for the reproduced study. After understanding the paper thoroughly, we interpolated that we want to maximize F1-score. For some phenotype, the Precision-recall were still off from the author's result since the different precision-recall combination can have the same F1 score.

Some of the things which were in our favor:

- We were able to reproduce the study in minimum iteration. This was the reward for our awareness of the various hyper-parameters involved and for trying to be as close to the original study.
- Most of our code was structured into specific methods and hence plugging in different architectures was feasible.

5.1 Recommendations for reproducibility

- Having the correct chart time in the annotation provided by the author would have helped join the data accurately and would save us from merging the discharge summary making our experiment closer to what they did.
- Given the number of labeled datasets is quite low (1610 data points), hence providing the train, validation, and test set used would be better to reduce the variance introduced by data.

6 Conclusion

We were able to reproduce the original study and perform various ablation studies to understand the importance of components in the architecture.

References

[1] S. Gehrmann, F. Dernoncourt, Y. Li, E. T. Carlson, J. T. Wu, J. Welt, J. Foote Jr, E. T. Moseley, D. W. Grant, P. D. Tyler *et al.*, "Comparing deep learning and concept extraction based methods

for patient phenotyping from clinical narratives," *PloS one*, vol. 13, no. 2, p. e0192360, 2018. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5813927/#pone.0192360.s001

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[3] S. Gehrmann, "Repository for comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives." [Online]. Available: https://github.com/sebastianGehrmann/phenotyping