Automated ICD-9-CM Medical Coding of Diabetic Patient's Clinical Reports

Sandeep Kumar and Nisarg Mistry
{kumar64, nmistry2}@illinois.edu

Group ID: 100, Paper ID: 137
Presentation link: https://www.youtube.com
Code link: https://github.com/MLwithSandy/CS598_DLH_Project

1 Introduction

Most of the patient health data, which is written by health professionals is stored as Electronic Health Records (EHR). All the data about the patient's state/health across time are extremely important for good diagnosis and treatment of the patient underlying health condition. These EHR records are also used by other players in health industry e.g. insurance companies uses various diagnosis and procedure codes in EHR records for cost reimbursements. The diagnosis and procedure codes used in the EHR records follows ICD-9-CM or ICD-10-CM standard cods. If these codes are not assigned correctly in EHR or there are missing codes, it could cause legal, economic implications for the patient, institutions and the insurance companies, involved in the health care chain. Assigning and correcting these codes are done manually by specialized medical personnel, called medical coders, and is estimated to cost about 25 dollars billion per year in the United States.

In this paper, we try to replicate the work done by Vitor Pereira et. el. (Pereira et al., 2018) and reproduce the baseline for a system to automate the assignment of ICD-9-CM codes to clinical reports. We use classification models to approximate the medical report's text content and try to map it to appropriate medical assigned codes. Following the approach taken by authors of original paper (Pereira et al., 2018), we use deep learning models and neural networks to perform ICD-9-CM code assignments for medical notes from EHR records. We work with a subset of MIMIC-III dataset - notes and diagnosis codes for those patients, who have been diagnosed with diabetes.

2 Scope of reproducibility

The paper (Pereira et al., 2018) uses two different type of classifiers for multi-level classification - Bag of Tricks (BoT) and Convolution Neural Network (CNN) and claims the performance of CNN is better than BoT as BoT classifiers does not care about word order and also multi-word expressions are not established.

2.1 Addressed claims from the original paper

In this paper, we reproduce the baseline established in original paper (Pereira et al., 2018) to test following claims

- CNN Baseline has higher Precision, Recall and F1 as compared to BoT Baseline
- CNN 3-Conv1D has higher Precision, Recall and F1 than CNN Baseline
- Precision, Recall and F1 for Rolled up ICD-9-CM code assignment is always higher than Regular ICD-9-CM code assignment

3 Methodology

To reproduce the baseline established in the original paper(Pereira et al., 2018), we are following the text description and using the same test dataset - MIMIC III used by authors of original paper (Pereira et al., 2018). For computation, we are planning to utilize Google Colab and leverage GPUs/TPUs for CNN model training. For data preprocessing, we are using personal laptop with 2 CPU CORE, 16GB RAM.

3.1 Model descriptions

The original paper uses two types of models - Bag of Tricks(BoT) neural network for fast model training and Convolutional Neural Networks (CNN). The paper tries to experiment with two types of CNN models. One with a 3-stage computation architecture which takes the Word2Vec embeddings as input. The second CNN architecture used is a

3 parallel convolutional neural network architecture with different sizes of filters being used for performing convolutions. The output of these 3 parallel CNNs are concatenated before passing it through the dense layer for predictions.

3.2 Data descriptions

In the paper, we use the Medical Information Mart for Intensive Care III (MIMIC-III) dataset which is a large, freely available database of anonymized health-related data associated with patients of the intensive care unit (ICU) of the Beth Israel Deaconess Medical Center between 2001 and 2012 (Joulin et al., 2016). MIMIC-III is composed of several tables which host several types of data from admissions and transfers to patient information and test results. For the purposes of this thesis only two are used: the notes table (NOTEEVENTS) which contains reports input by clinicians and nurses into the EHR at each patient admission and the diagnosis table (DIAGNOSIS_ICD) which contains ICD-9-CM codes, associated with each admission, generated for billing purposes at the end of each patient's hospital stay.

3.3 Hyperparameters

We try to follow the details as in original paper (Pereira et al., 2018) to establish the baseline. In case of missing information, we plan to go with defaults or make a suitable assumption based on test data.

3.4 Implementation

For the implementation, we are implementing the code from scratch. There is no code available for reference so we would be following the details, mentioned in the original paper(Pereira et al., 2018) for the implementation.

3.4.1 Data Preprocessing

Records for patients who are already diagnosed with diabetes (ICD-9-CM code: 250.xx) are selected first from MIMIC III datasets: DIAGNOSES_ICD and NOTEEVENTS. For classification, two different representation of ICD-9-CM codes are considered - regular one with three to five digits and rolled up version with first three digits. After that following four steps are executed to cleanse the text and tokenize the report.

1. Punctuation characters, except apostrophe, is replaced with whitespace
2. Digits are replace by character d
3. All characters are converted to lowercase
4. Reports are tokenized, splitting at whitespace

Post data cleansing and tokenization, simple vocabulary reduction is done for words with frequency less than 5, assuming that these words are misspellings and are replaced with words from the vocabulary (consisting of words from all reports), with least Levenshtein distance. At the end, all reports which consists of less than 9 tokens or having more than 2200 tokens are eliminated.

3.4.2 Embedding

For text classification task, the tokens needs to be converted to numerical, matrix representation of the tokens using word embeddings. Word2Vec's skip-gram neural network model was trained using all tokens from the EHR records notes and 300 dimensional word embedding was generated. The implementation uses gensim library for Word2Vec model (Mikolov et al., 2013) implementation.

3.4.3 Bag Of Tricks

This model is a simple linear model based on Armand et al.'s Bag of Tricks (BoT) (Joulin et al., 2016). This model is composed of an embedding layer that transforms the input dataset tokens into high dimension word embeddings. The output from this layer is then averaged into a fixed dimension vector before it is passed as input in the dense/fully-connected layer. The output of this layer depends on the task and the number of output labels required for that particular task. The output of the dense layer is passed through a sigmoid activation function squashing the input in the range of [0-1] so that the output is a Bernoulli probability distribution. We use the Adam optimizer which enables the adaptive tuning of the classifiers hyperparameters during training and we use batches of 32. The classifier uses no other regularization except for early stopping with a minimum delta of 0.0001 and a patience of 2 epochs.

3.4.4 Convolutional Neural Networks

This model is a Convolutional Neural Network architecture which is similar to Bag of Tricks model discussed earlier. Here the embeddings are passed as input into a CNN 3 stage computation architecture. We pass the embeddings through a one-dimensional convolution layer with filter size of

250 and kernel size of 3 outputs a set of linear activations. Next, each linear activation is run through a (non-linear) rectified linear unit(ReLU). The final layer is a max pooling layer which helps to make the output representation invariant to small translations in the input.

3.4.5 Three layered Convolutional Neural Networks

This model is similar to the CNN architecture discussed above. However, in this architecture, we use 3 parallel CNNs with different filter sizes. In this architecture, the 3 CNNs use different filter sizes for performing varied sets of convolutions. Before passing the CNN output through dense/fully-connected layer, we concatenate the results from all 3 CNNs and pass it into the dense layer. All the hyper parameters used are the same as the above mentioned CNN.

3.5 Computational requirements

We plan to utilize our personal laptop with following configuration: 2 CPU Cores and 16 GB memory. In case, higher configuration of computation resources are required, we plan to use Google Colab with 16 CPU/GPU Cores and 64 GB memory.

4 Results

For multi-level classification of text, preprocessing of text data and word embeddings are required. So far, we have spent significant amount of effort on preprocessing of text data and generation of word embeddings.

The result of data preprocessing is very close to the result achieved in original paper(Pereira et al., 2018).

	Reprod.	Original
Num. of used records	399629	399623
Num. of regular labels	4103	4006
Num. of rolled up labels	781	779
Num. of unique tokens	53304	53229
Avg. num. of tokens per report	309.62	309.06

4.1 Result 1

Implementation of BoT and CNN models are completed and we are in progress to test it with MIMIC III test data.

Analysis: Based on our understanding of the original paper, Bag of Tricks (BoT) performance should be inferior as compared to CNN model since BoT does not care about word sequences and relation between words in the EHR records.

As part of the experiment, we plan to reproduce following result:

- BoT BaseLine (Precision, Recall, F1): (66.25, 8.61, 15.24)
- CNN Baseline (Precision, Recall, F1): (73.97, 25.88, 38.13)

4.2 Result 2

Implementation of CNN 3 layered model is still in progress. As part of the experiment, we plan to reproduce following result:

- CNN Baseline (Precision, Recall, F1): (73.97, 25.88, 38.13)
- CNN 3-Conv1D (Precision, Recall, F1): (76.07, 31.46, 44.51)

Analysis: Based on our understanding of the original paper, CNN 3 layered model should perform better than CNN baseline model as we are performing three varied convolutions on the input word embedding, which helps in capturing additional information from the EHR records.

4.3 Result 3

Implementation in progress

Analysis: Based on our understanding of the original paper, Prediction of Rolled up ICD-9-CM code has higher precision, recall, F1 as compare to regular codes as more test data is available for prediction of rolled-up code, which helps in creating a generalized model.

4.4 Additional results not present in the original paper

TODO

5 Discussion

TODO

5.1 What was easy

TODO

5.2 What was difficult

TODO

5.3 Recommendations for reproducibility

TODO

6 Communication with original authors

We contacted Vitor Pereira, author of original paper(Pereira et al., 2018) over email. His initial response was delivered to junk mailbox and code (python files), shared as attachment, was removed by Illinois email set-up. We are waiting for feedback for subsequent email communication, requesting codes via github or gmail id.

References

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2016. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Vitor Pereira, Sérgio Matos, and José Luís Oliveira. 2018. Automated icd-9-cm medical coding of diabetic patient's clinical reports. In *Proceedings of the First International Conference on Data Science, E-Learning and Information Systems*, DATA '18, New York, NY, USA. Association for Computing Machinery.