

Classifying medical notes into standard disease codes

Zenobia Liendo

University of California
Berkeley, CA
zenla@berkeley.edu

Guillaume De Roo

University of California
Berkeley, CA
gderoo@berkeley.edu

Amitabha Karmakar

University of California
Berkeley, CA
amitkarmakar@berkeley.edu

Abstract

We investigate the automatic classification of patient discharge notes into standard disease labels. We find that Convolutional Neural Networks with Attention outperform previous algorithms used in this task, and suggest further areas for improvement.

1 Introduction

Electronic Health Records (EHRs) have grown significantly over the years and now include an unprecedented amount and variety of patient information, including demographics, vital sign measurements, laboratory test results, prescriptions, procedures performed, digitized notes, imaging reports, mortality etc. They usually contain both structured data (e.g. admission dates) as well as unstructured data (e.g. notes written by doctors).

Provided it can be processed, the information in these records - especially the unstructured data - holds the promise of new medical insights and improved medical care, such as faster detection of epidemics, identification of symptoms, personalized treatment, or a more detailed understanding of treatment outcomes.

One such gain is a more automated and accurate way to report diseases. Since 1967, the World Health Organization (WHO) has developed an International Classification of Diseases (ICD) to “monitor the incidence and prevalence of diseases, observe reimbursements and resource allocation trends, and keep track of safety and quality guidelines”¹. Currently this ICD labeling is done manually by administrative personnel based on definitions and is subject to interpretation and errors².

In this paper, we focus our efforts on the automatic labeling of discharge notes from the

MIMIC³ Database into ICD codes. This public database of EHRs contains data points on about 41,000 patients from an intensive care units between 2001 and 2012, including notes on close to 53,000 admissions. MIMIC has already been proven valuable for efforts similar to ours, which will make comparisons more accurate.

2 Background

The problem of assigning ICD codes automatically to discharge summaries has previously been studied. Among the most recent research publications, we can distinguish two types of efforts: papers trying to predict the ICD codes in all their complexity, and those more numerous who focus on a smaller domain.

Full ICD codes: Perotte et al. (2014) used the MIMIC II dataset to predict the original ICD codes. They experimented with two approaches: one that treats each ICD9 code independently of each other (flat classifier), and one that leverages the hierarchical nature of ICD9 codes into its modeling (hierarchy-based classifier). They used a novel evaluation metrics, which reflected the distances among source ICD9 tree and predicted codes and their locations in the ICD9 tree. They found that the hierarchy-based classifier outperformed the flat classifier.

Simplified ICD codes: Other researchers focused their efforts on a smaller number of ICD codes, and found their best results using Convolutional Neural Networks (CNNs). Gerhman et al. (2017) relabeled 1.6K clinical notes from MIMIC III using their own 10 labels. They find that CNNs outperform other approaches based on n-gram models, and Natural Entity Recognition (NER).⁴

¹<http://www.who.int/classifications/icd/en/>

²See recent articles on opioid overdose statistics in the US <https://tinyurl.com/y8zc3xcq>

³Medical Information Mart for Intensive Care <https://mimic.physionet.org>

⁴Using cTAKES <http://ctakes.apache.org/>

Our Approach: In this paper, we focused on improving techniques applied to the simplified ICD code problem. While CNNs seem well suited, some characteristics of discharge notes raise possibilities for other approaches.

Medical notes describe a temporal sequence of events and tests, to which CNNs are oblivious, on the contrary to Long Short-Term Memory (LSTM) models. Additionally, notes are long, with an average length of around 1500 words. Because of this large context, we also explored Attention models which do not seem to have been applied to this problem domain before.

Last, we briefly investigated how the two approaches (full and simplified ICD codes) could be reunited by adapting the training metric.

3 Methods

We focused on the classification of hospital admission discharge notes into ICD-9 codes, using the MIMIC III database (Johnson et al., 2016) for comparison purposes.

We can broadly break our approach to this multi-label multi-class problem into the steps detailed below: output labeling, input preprocessing, training and output metrics, and algorithms.

3.1 Output labeling

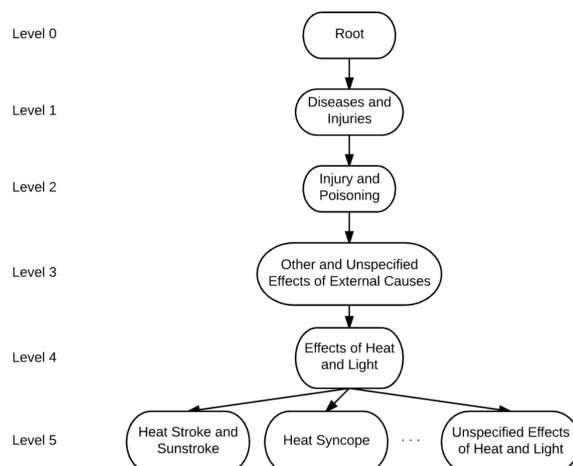
The ICD-9 nomenclature applied by MIMIC III contains about 14,000 numerical codes representing all possible diagnoses and procedures⁵. Out of those 14,000, 5,932 distinct codes are used to describe the 52,696 hospital admissions of the database, with 1,112 codes appearing only once.

This creates an issue for classification algorithms since many codes would need to be predicted with few or no example in the training set. Fortunately, the ICD codes are organized in a hierarchical tree, see Figure 1.

As a result, we identified 3 mains methods to deal with the high number of classes:

- Restrict the labels to the most common Level 5 codes, method used by some project reports.⁷ We start by selecting the 20 most common codes (see Figure 2)
- Relabel all codes into a smaller class of codes. This approach was done manually by

Figure 1: Sample ICD9 path⁶



(Gehrmann et al., 2017). Here, we take advantage of the ICD hierarchy, and simply relabel notes into the 17 nodes of depth 1.⁸

- A third possible approach - to explore in further work - would be to keep all labels, but use a "hierarchical metric", i.e. discounting errors if labels are in the same ICD branch.

Intuitively, we can expect the second approach to perform better, since i) the codes represent very different realities, whereas common codes may be related, and ii) the distribution is less balanced. An additional benefit is that we have access to the full dataset for training (53K) instead of just a subset if we had re annotated the dataset manually or if we take the most common codes (46k).

The third untested approach would allow to keep all codes intact, and hence be more precise in the labeling.

3.2 Note preprocessing

The database presents multiple clinical notes categories including things like Radiology, Nutrition, Pharmacy, or Social Work. Here, we focus on Discharge Summaries,⁹ which already provide a synthesis of different aspects.

To process those notes, we go through relatively common steps that we summarize briefly here: we put words in lower case, remove most special characters, separate contractions, canonize numbers, and tokenized the resulting words.

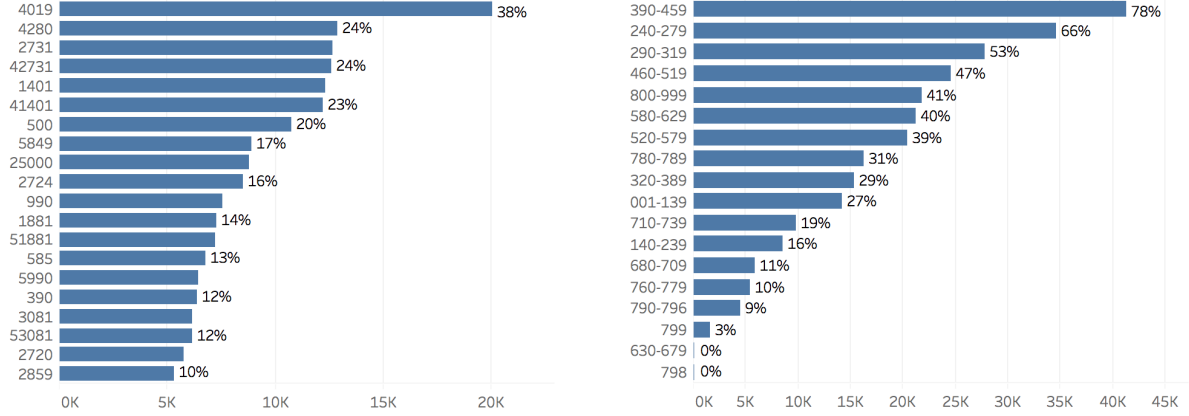
⁸Excluding 798 which appears only once

⁹There are 2 types of discharge summaries, reports and addendum, we focused on the reports

⁵ICD-10 (current version) has around 68,000 labels

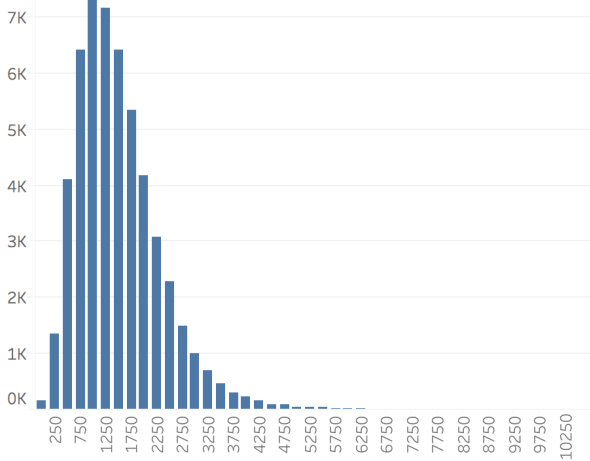
⁷<https://cs224d.stanford.edu/reports/lukelefebure.pdf>

Figure 2: Penetration of top 20 Level 5 codes (left) and all Level 1 codes (right)



The results is a vectorized set of notes, which can reach 10,924 words. Since some of our algorithms require a fixed length input, we truncate and pad the notes so that the output has a length of 5,000 words. This is done without loss of generality, since 99.5% of the notes meet this criteria.

Figure 3: Original distribution of note length



3.3 Embedding

Unfortunately, even after the previous steps, the wording is still not standardized. Like some unpublished papers, we can see at least 13 ways that write hypercholesterolemia, for instance.¹⁰

One way to solve this issue would be to use Named Entity Recognition (NER). Some implementations exist which are tailored to the medical realm, such as Apache cTAKES or MetaMap. However, previous papers (Gehrmann et al., 2017) find that embeddings perform better, trusting embeddings' ability to make "misspellings, syn-

¹⁰<https://cs224d.stanford.edu/reports/priyanka.pdf>

onyms and abbreviations of an original word learn similar embeddings". Therefore we used trainable embeddings, sometimes pre-trained with the Glove algorithm on Wiki¹¹ or on the MIMIC notes to account for the vocabulary specificity.

Note that some papers such as (Perotte et al., 2014) use TF-IDF, either to restrict the original vocabulary size or to transform notes into continuous components. Here since we use embeddings of size 100, we can keep our original vocabulary of 60,619 with limited impact to our calculation time.

3.4 Training Loss Function

For multi-label classifications like this one, an approach is to convert the problem into single binary classification tasks. This would not work for ICD-9 codes since the ones assigned to a clinical-note may not be independent (some medical conditions are correlated).

An early procedure for multi-label classification using NNs was BP-MLL which uses a novel pairwise ranking loss function for training (Zhang et al., 2006), but later research found that cross entropy produces better results (Nam et al., 2013). In this work, we use the latter.

3.5 Algorithms

For all of our models, we used standard software and evaluation methods.¹²

¹¹<https://nlp.stanford.edu/projects/glove/>

¹²Data is split between training (70 %), validation (15%) and test (15%) sets. Models were implemented using TensorFlow and Keras. Training optimizer used was Adam. Our models are using L2 regularizations and Dropouts, which proved its efficiency. Default parameters were used.

3.5.1 Baseline and Linear Models

Our Baseline model simply predicts the 4 most common ICD-9 codes for each clinical note. This performs better than more complicated alternatives, for example Gehrmann et al. (2017) used a 3-gram Logistic Regression with relatively poor results (Table 1).

3.5.2 CNN

CNNs have been used for image multi-label classification. Although the invariances are different between an image and a text, this sounds similar to our problem.

This work implements a CNN for text classification replicating the architecture presented by Kim (2014) and based on hyper parameters tested by Gehrmann et al. (2017).

The CNN model has one layer of convolution which used 4 different sized windows. Each window takes 2,3,4 or 5 words and applies 100 filters, encompassing the full embedding size.

We use this model to classify into Level 5 ICD-9 codes and first-level ICD-9 codes in the hierarchy.

3.5.3 LSTM

According to Yin et al. (2017) the state-of-the-art on many NLP tasks often switches between CNNs and RNNs (LSTM in this case), his paper lists different past studies where sometimes a CNN performs better and other times a LSTM.

Hence we implement a LSTM model to see if some of the discharge note features (e.g. temporal sequence) make it a better candidate. Since we have a relatively small file (56K records), we start with a single layer LSTM to keep the number of parameters low. We didn't find published papers regarding classifying clinical notes using LSTM, however we did find a report on the web.¹³

3.5.4 Attention

As explored in Section 3.2, the average length of discharge clinical notes is 1639 words. The text to classify may be too long for a LSTM or CNN to remember all relevant information.

Raffel et al. (2016) displayed better performance in many NLP tasks on long text using Attention. Here, we seek to emulate his results by implementing algorithms based on the formulas presented in (Raffel et al., 2016) and Yang et al. (2016).

LSTM with Attention: The LSTM cell returns not only the last hidden state but all the intermediate ones that are then sent to the attention layer which creates a new vector representing the clinical note for the output layer classification.

CNN with Attention: The MaxPooling element in the CNN network is replaced by the Attention layer in order to create a vector representing all relevant information and not only taking in account max values. A model like this one is mentioned in Yin et al. (2016)

Hierarchical Attention: This model was implemented based on Yang et al. (2016) which specifically targets document classifications. It has two levels of attention mechanisms, the first one creates vectors that represent each sentence, using attention mechanism across words; and the second level creates a vector that represent the document using attention mechanisms across sentences. Yang et al. (2016) uses Bidirectional GRUs while we use LSTMs for a fair comparison with the flat LSTM models.

3.6 Threshold Calibration

The resulting vector from the neural network may be interpreted as a probability of the individual ICD-9 codes (each cell has a value 0-1, but does not sum to 1). To complete the prediction, we must convert the vector to binary values.

There are several methods for selecting a Threshold (Zhang et al., 2016). We used a constant threshold maximizing the overall F1-score. In future work, we could explore methods building a (linear) model on top of the intermediate vector.

3.7 Performance Metrics

We use the F1 metric on the validation data to evaluate performance in all models and compare results with previous work on classifying MIMIC clinical notes and text classification in general.¹⁴

4 Results and Discussion

4.1 Comparing CNN with previous work

In order to compare F1 performance results with the CNN model built by Gehrmann et al. (2017), we took into consideration the dataset size and number of classes.

¹⁴For multi-label classifications, sklearn offers several options, we used the F1 'micro' option which calculates global counts for true positives, false negatives and false positives.

¹³<https://cs224d.stanford.edu/reports/priyanka.pdf>

Gehrmann’s re-labeling approach is similar to our relabeling using the first-level ICD-9 codes in the ICD code hierarchy. Even though we have access to 52.6K records, we use a subset to relate to the 1.6K records used by Gehrmann et al. (2017). Since we have 17 classes, 7 more than the ones used by Gehrmann , we run our model with a dataset of 5K records.

Our CNN obtains similar result to Gehrmann et al., with a F1 score of 76.2%, compare to their F1 score of 76% (see Table 1).

Source	Labels	Methods	Rec	F1
Gehrmann et al., 2017	10 own labels	LR 3-gram	1.6K	34.6
Gehrmann et al., 2017	10 own labels	CNN	1.6K	76
This Paper	17 ICD-9	CNN	5K	76.2

Table 1: Classification of MIMIC clinical notes into labels representing high level phenotype categories (20 epochs for both CNN models)

4.2 Testing CNN, LSTM and Attention

To improve on this initial result, we ran experiments with the different models to identify the two more promising. These experiments run with a 5K notes, the 17 first level ICD-9 codes, using 5 epochs. The results are presented in Table 2

We tested LSTMs with and without attention mechanisms, CNN with and without attention mechanisms and a Hierarchical LSTM model with Attention layers.

From the results in Table 2 we can see that CNN models do perform better than LSTM on classifying the MIMIC medical notes.

We can also see that there is a significant improvement on F1 scores when applying attention mechanism to LSTM and CNN models. The LSTM with Attention model outperforms the standard LSTM by 2.4% and the CNN with Attention model outperforms the standard CNN model by 3.8%.

On the other hand, the Hierarchical LSTM with Attention mechanisms had only a small increase (0.6%) in performance results on regards to the Flat LSTM with Attention. This is smaller than we expected based on similar classification tasks

Source	Methods	Recs	F1
This Paper	LSTM	5k	64.6
This Paper	LSTM-Attention	5k	67
This Paper	Hierarchical LSTM-Attention	5k	67.6
This Paper	CNN	5k	69
This Paper	CNN-Attention	5k	72.8

Table 2: Classification of MIMIC clinical notes into Level 1 ICD-9 Codes. Evaluation with 17 classes, 5k records, 5 epochs

by Yang et al. (2016), where a difference of 3% is reported, but on larger datasets. This model has twice the number of parameters than the flat models, which would impact performance for relatively small files like the one we are using, this could be a reason for just a small improvement in the f1 score. We also tried GRUs instead of LSTMs to compare with Yang et al. (2016) results and the difference was still the same.

Another possible reason our Hierarchical model is not performing much better is the tokenization of sentences. The model bases its predictions on the results on each sentence, and if the sentences are not identified correctly in the first place, then the rest of the model will not perform well. We did inspect suspicious long sentences which were not incorrect, they were lab reports. We would inspect closely the sentence tokenization process in further work.

The two **most promising models are CNN and CNN with Attention**, even the standard CNN model outperforms the Hierarchical model.

CNN models could be seen as hierarchical: the convolutional sliding windows create segments of the document (like sentences do) and they are collapsed into vectors representing a higher level of abstraction. In as sense CNN are finding the best segments in the document regardless of sentences separations. This may explain why the CNN models are getting a better performance than the Hierarchical models.

4.3 CNN performance with full data set

Here we show results from running the CNN models with the full data set.

First we classify clinical notes into the 20 most common Level 5 ICD-9 codes for comparison purposes: we can see that our model outperformed previous work (see Table 3).

Source	Methods	N. Rec	F1
Perotte et al., 2014	Hierarchical SVM (all codes)	22K	39.5
Previous Project Reports ¹⁵	LSTM	32K	41.6
This paper	Baseline	46K	35
This paper	CNN	46K	72.4

Table 3: Classification of MIMIC clinical notes into most common Level 5 ICD-9 Codes

To go further, we trained both CNN Models to classify clinical notes into Level 1 ICD-9 codes in the hierarchy (see Table 4).

Source	Methods	Recs	F1
This Paper	Baseline	52.6K	53
This Paper	CNN	52.6K	79.7
This Paper	CNN w/ Attention	52.6K	78.2

Table 4: Classification of MIMIC clinical notes into 17 Level 1 ICD-9 Codes

As anticipated earlier, the plain CNN model executed with the 52.6K records got a F1-score of 79.7%, outperforming any other model in previous work, due to i) a larger dataset, ii) better separated labels, and iii) a more imbalanced label distribution (see Section 3.1).

However at this stage, the CNN ATT model still overfits: even though it had the highest score during the experimental runs with 5K records and 5 epochs, it didn't reach the best f1-score when running it with the full data set. Further work would explore hyper-parameters tuning and evaluating the number of parameters to attempt undoing the over fitting situation.

We believe the CNN models can still be improved by inspecting in more detail cases where the model predicted a false positive or false negative, and working on hyper-parameters. This would be one of the first tasks to do in further work regarding these models.

4.4 Pre-trained Embeddings

We used trainable embeddings, as described in Section 3.3. Our two attempts to initialize them with pre-trained values were unsuccessful.

Using the Wiki Glove pre-trained embeddings led to a minor decrease in performance (about 0.001%) compared to an empty embedding matrix, which could be expected since Medical clinical notes have a vocabulary that differs from most Wiki pages. In fact, half of our vocabulary was not found on the Wiki Glove pre-trained embeddings.

We then created our own pre-trained embeddings using the Glove algorithm on all the MIMIC discharge notes. The result was a small performance improvement of 0.01%.

As part of future work, we think that using pre-trained embeddings on millions of clinical notes would improve the performance of models processing clinical notes, this is an example of such type of work ¹⁶.

5 Conclusions and outlook

In this paper, we tested several alternative approaches for classifying ICD-9 codes.

We showed that our a CNN models outperform significantly the F1 scores reported by previous work on Level 1 or Level 5 codes, while LSTMs and Hierarchical model displayed lower performance.

However for the problem of automatic labeling to be solved, models need to increase both in performance and in the precision of the codes that they allocate. Our results highlight several areas to further that goal:

- optimization of CNN model with Attention, given promising results on small datasets
- better adapt embeddings to clinical notes
- broaden the number of ICD codes, gradually or by adapting the training metric

Finally we note that ICD codes are associated with a textual definition which could be directly compared with the clinical notes themselves.

References

Sebastian Gehrmann, Franck Dernoncourt, Yeran Li, Eric T Carlson, Joy T Wu, Jonathan Welt, John Foote Jr., Edward Moseley, David W Grant, Patrick D Tyler, Leo Anthony Celi. 2017. *A Comparison of Rule-Based and Deep Learning Models for Patient Phenotyping*. MIT Critical Data, Laboratory for Computational Physiology, Harvard John

¹⁵<https://cs224d.stanford.edu/reports/priyanka.pdf>

¹⁶<https://web.stanford.edu/class/cs224n/reports/2744372.pdf>

A. Paulson School of Engineering and Applied Sciences, Massachusetts Institute of Technology, Harvard T.H. Chan School of Public Health, Philips Research North America, Beth Israel Deaconess Medical Center, Massachusetts General Hospital, Tufts University School of Medicine, University of Massachusetts, Washington University School of Medicine

Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, Nomie Elhadad. 2014. *Diagnosis code assignment: models and evaluation metrics*. Department of Biomedical Informatics, Columbia; University, New York, New York, USA; New York Presbyterian Hospital, New York; Department of Engineering, University of Oxford, Oxford, UK

J. Nam, J. Kim, E. Loza Menca, I. Gurevych, and J. Furnkranz. 2013. *Large-scale Multi-label Text Classification - Revisiting Neural Networks*. ArXiv e-prints.

M.L. Zhang and Z.H. Zhou. *Multi-label neural networks with applications to functional genomics and text categorization*. 2006. *IEEE Transactions on Knowledge and Data Engineering*, 18:13381351.

Yoon Kim. 2014. *Convolutional Neural Networks for Sentence Classification*. New York University.

Wenpeng Yin, Katharina Kann, Mo Yu and Hinrich Schutze *Comparative Study of CNN and RNN for Natural Language Processing*. 2017. CIS, LMU Munich, Germany, IBM Research, USA

Colin Raffel, Daniel P. W. Ellis *Feed-Forward Networks With Attention can solve some long-term memory problems*. . 2016. *Workshop track - ICLR 2016*

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, Eduard Hovy *Hierarchical Attention Networks for Document Classification*. 2016. Carnegie Mellon University, Microsoft Research, Redmond.

Wenpeng Yin, Hinrich Schutze, Bing Xiang, Bowen Zhou *ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs*. 2016. Carnegie Mellon University, Microsoft Research, Redmond.

Min-Ling Zhang, Zhi-Hua Zhou *A Review on Multi-Label Learning Algorithms*. 2014. *IEEE Transactions on knowledge and data engineering*, VOL. 26, NO. 8

Johnson AEW, Pollard TJ, Shen L, Lehman L, Feng M, Ghassemi M, Moody B, Szolovits P, Celi LA, and Mark RG. *MIMIC-III, a freely accessible critical care database*. 2016. *Scientific Data* (2016). DOI: 10.1038/sdata.2016.35.