A Graph Enhanced Label Attention Model for ICD Coding from Clinical Text

Anonymous ACL submission

Abstract

Medical code assignment from clinical texts is a crucial task in the healthcare industry. Clinical texts are typically very long sequences and the number of possible labels are large, making this task quite challenging. Recent work applies deep neural network models to encode the medical notes and assign medical codes to clinical documents. Some works use effective attention mechanisms to construct label-specific document representations and show promising results. In this paper, we propose a new attention mechanism, GE-LAAT (graph enhanced label attention), which utilizes code graphs to learn robust representation vectors for medical codes and improve upon the state of the art models. Experiments on the MIMIC-III dataset are conducted to show the effectiveness of our proposed model.

1 Introduction

Medical notes are text documents written by clinicians during patient encounters. These notes are usually accompanied by a set of codes from the International Classification of Diseases (ICD), which present a standardized way of indicating diagnoses and procedures that were performed during the encounter. The codes are then used for different purposes such as billing or predictive modeling of patient state (Choi et al., 2016; Ranganath et al., 2015; Denny et al., 2010; Avati et al., 2017). Manual coding by a human coder can be very challenging due to many reasons. First, the label space is very high-dimensional, with over 15,000 codes in the ICD-9 taxonomy. Second, a typical text is very lengthy, includes irrelevant information, misspellings and non-standard abbreviations, and a large medical vocabulary (Birman-Deych et al., 2005). Hence, there is a need for an accurate automated coding system to overcome these issues. In this paper, we improve upon the state of the model LAAT (Vu et al., 2020) by proposing a graph enhanced label attention mechanism and a new approach to learn code representations and show the effectiveness of our approach via experiments.

2 Related Work

CNN (Kim, 2014) uses pretrained word vectors with 1D convolution and max pooling for text classification. CAML (Mullenbach et al., 2018) integrates CNNs and a label-wise attention mechanism to learn rich representations. It has a variant called DR-CAML that uses ICD code descriptions to regularized the loss function. MultiResCNN (Li and Yu, 2020) combines residual learning (He et al., 2016) and multiple channels concatenation with different convolutional filters. HyperCore (Cao et al., 2020) utilizes hyperbolic embedding and co-graph representation with code hierarchy. MSATT-KG (Xie et al., 2019) contains a densely connected convolutional neural network which can produce variable n-gram features and a multi-scale feature attention to adaptively select multi-scale features. The graph convolutional neural network (Kipf and Welling, 2017) is also employed to capture the hierarchical relationships among medical codes. Gated-CNN-NCI (Ji et al., 2020) uses a gated CNN along with a note-code interaction module which uses a graph message passing mechanism to capture the dependency between notes and codes. LAAT (Vu et al., 2020) uses a bidirectional LSTM, followed by an attention mechanism and obtains the best results among all state of the art methods. It has a variant called JointLAAT which uses the hierarchical structure among the codes. In this paper, we focus on improving LAAT, by designing a more effective attention mechanism.

3 Methodology

3.1 Problem Definition

A clinical note $\mathcal{X}$ with n words is represented as $\mathcal{X} = [w_1, w_2, ..., w_n]$ where each w_i repre-

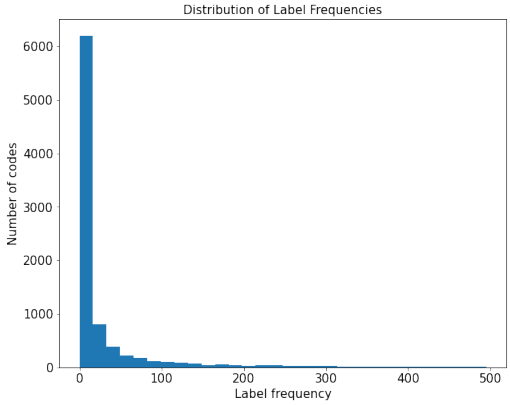

Figure 1: The distribution of label frequencies in the training set. The x-axis shows the label frequency and the y-axis shows the number of different codes that are observed with that frequency.

sents a word. The set of all possible codes is $L = \{c_1, c_2, .., c_{|L|}\}$ and $|L|$ is the total number of labels in the dataset. The goal is to find a mapping $\mathcal{F} : \mathcal{X} \mapsto \mathbf{y}$ such that

$$\mathbf{y} = \mathcal{F}(w_1, w_2, ..., w_n; \mathcal{D}) \quad (1)$$

where $\mathcal{D}$ is available side information and $\mathbf{y} \in \mathbb{R}^{|L|}$ is multi-hot indicator vector with $\mathbf{y}_i = 1$ if note $\mathcal{X}$ contains code $\mathbf{c_i}$.

3.2 Node2vec

Node2vec (Grover and Leskovec, 2016) is a framework for learning continuous feature representations for nodes in networks. The model learns a mapping of nodes to a low-dimensional space of features that maximizes the likelihood of preserving network neighborhoods of nodes. A flexible notion of a node's network neighborhood is defined and a diverse set of neighborhoods are explored using a biased random walk procedure. Formally, the the following objective function is optimized:

$$\max_f \sum_{u \in V} \log \Pr\left(N_S(u)|f(u)\right) \quad (2)$$

which maximizes the log-probability of observing a network neighborhood $N_S(u)$ for a node u conditioned on its feature representation, given by f under sampling strategy S. Assuming that the likelihood of observing a neighborhood node is independent of observing any other neighborhood node given the feature representation of the source, the likelihood as factorized as:

$$\Pr\left(N_S(u)|f(u)\right) = \prod_{n_i \in N_S(u)} \Pr\left(n_i|f(u)\right) \quad (3)$$

where the conditional likelihood of every source-neighborhood node pair is modeled as a softmax unit parametrized by a dot product of their features:

$$\Pr\left(n_i|f(u)\right) = \frac{\exp\left(f(n_i) \cdot f(u)\right)}{\sum_{v_i \in V} \exp\left(f(v) \cdot f(u)\right)} \quad (4)$$

We use the node2vec model to generate the pretrained embedding matrix $\mathbf{P} \in \mathbb{R}^{d_p \times |L|}$, where d_p is the embedding dimensionality. We use two different graphs to generate pretrained embeddings, which are defined in the next section.

3.3 Graph Construction

The label distribution in the dataset is extremely unbalanced as shown in Figure 1, and it is difficult to detect the rare labels accurately with only a few samples. Thus, we leverage two different graphs to utilize: 1) hierarchical structure of codes. 2) cooccurence between codes.

3.3.1 Hierarchical Graph

The ICD codes are organized using a hierarchical structure. For example, the codes "Chemical burn of eyelids and periocular area" (940.0), "Other burns of eyelids and periocular area" (940.1) and "Alkaline chemical burn of cornea and conjunctival sac" (940.2), are all under a higher category called "Burn confined to eye and adnexa" (940). Similarly, "Burn confined to eye and adnexa" (940), "Burn confined to eye and adnexa" (941), "Burn of face head and neck" (942) and "Burn of trunk" (943) are all under a more higher category called "Burns" (940-949). This hierarchy naturally forms a tree structure, which is a connected acyclic undirected graph. Each node represents a code and there is an undirected edge between each node and its children. Note that the labels that we want to predict are the leaf nodes.

3.3.2 Cooccurence Graph

We build an alternative, data-driven graph in order to capture the co-occurence patterns between codes similar to (Cao et al., 2020) and (Chen et al., 2019). First, we construct the co-occurence matrix $\mathbf{M}$, where $\mathbf{M}_{ij}$ denotes the number of times code $\mathbf{c_i}$ and code $\mathbf{c_j}$ occur together in a medical note in the training set. We row normalize $\mathbf{M}$ and obtain $\hat{\mathbf{M}} = \mathbf{D^{-1}M}$, where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D_{ii}} = \sum_{j=1}^{j=|L|} \mathbf{M_{ij}}$. Some codes do not cooccur with any other code, hence they are disconnected from the graph. For a disconnected node, we add

an edge to all the other nodes with an equal weight, i.e., for a disconnected code $\mathbf{c_i}$, we set $\hat{\mathbf{M}}_{\mathbf{ij}} = 1/(|L|-1)$ for all $j \neq i$. Finally, we prune the graph by only keeping the top 3 neighbors with the highest edge weights for each node and dropping the remaining edges. For a disconnected node, we randomly pick 3 neighbors to keep.

3.4 GE-LAAT

In this section, we explain our model in detail. Note that we follow a similar approach to (Vu et al., 2020) and our main contribution comes from the design of the attention module. The architecture of our model is shown in Figure 2.

3.4.1 Embedding Layer

The embedding layer takes as input a clinical note $\mathcal{X} = [\mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_n}]$ with n words and outputs the corresponding pretrained embedding vectors $\mathbf{e_{w_1}}, \mathbf{e_{w_2}}, ..., \mathbf{e_{w_n}}$ for each word $\mathbf{w_i}$.

3.4.2 Bidirectional LSTM Layer

Given the input sequence $\mathbf{e_{w_1:w_n}}$ of vectors $\mathbf{e_{w_1}}, \mathbf{e_{w_2}}, ..., \mathbf{e_{w_n}}$, the bidirectional LSTM layer learns latent feature vectors representing each input word. We compute the hidden states of the LSTMs corresponding to the i^{th} word as:

$$\overrightarrow{\mathbf{h_i}} = \overrightarrow{LSTM}(\mathbf{e_{w_1:w_n}}) \quad (5)$$

$$\overleftarrow{\mathbf{h_i}} = \overleftarrow{LSTM}(\mathbf{e_{w_1:w_n}}) \quad (6)$$

where $\overrightarrow{LSTM}$ and $\overleftarrow{LSTM}$ denote forward and backward LSTMs respectively. The final representation vector h_i is formed as:

$$\mathbf{h_i} = \overrightarrow{\mathbf{h_i}} \oplus \overleftarrow{\mathbf{h_i}} \quad (7)$$

where $\oplus$ represents the concatenation operation. The dimensionality of hidden states $\overrightarrow{\mathbf{h_i}}$ and $\overleftarrow{\mathbf{h_i}}$ are both set as u, hence, $h_i \in \mathbb{R}^{2u}$. All h_i are concatenated to obtain the document matrix $\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, ..., \mathbf{h_n}] \in \mathbb{R}^{2u \times n}$.

3.4.3 Graph Enhanced Label Attention Layer

This layer transforms $\mathbf{H}$ into label specific vectors using our graph enhanced attention mechanism. The mechanism takes $\mathbf{H}$ as input and outputs $|\mathbf{L}|$ label-specific vectors representing the input document.

$$\mathbf{Z} = tanh(\mathbf{W_1 H}) \quad (8)$$

$$\mathbf{U} = leakyReLU(\mathbf{W_2 P} + \mathbf{b}) \quad (9)$$

$$\mathbf{A} = softmax(\mathbf{U}^\top \mathbf{Z}) \quad (10)$$

Here, $\mathbf{P} \in \mathbb{R}^{d_p \times |L|}$ is the pretrained node2vec embedding matrix (generated using one of hierarchical/co-occurence graph) where each column of $\mathbf{P}$ represents the pretrained node2vec embedding for a single code c_i. $\mathbf{W_1} \in \mathbb{R}^{d_a \times 2u}$ and $\mathbf{W_2} \in \mathbb{R}^{d_a \times d_p}$ are trainable weight matrices, $\mathbf{b} \in \mathbb{R}^{d_a}$ and d_a is a hyperparameter. We transform the document representation $\mathbf{H}$ and the original node2vec vectors $\mathbf{P}$ using $\mathbf{W_1}$ and $\mathbf{W_2}$ respectively in order project them into the same space. $\mathbf{U} \in \mathbb{R}^{d_a \times |L|}$ and $\mathbf{Z} \in \mathbb{R}^{d_a \times n}$ are multiplied and a softmax is applied at the row level to obtain the label-specific weight matrix $\mathbf{A} \in \mathbb{R}^{|L| \times n}$, where each element $\mathbf{A_{ij}}$ of $\mathbf{A}$ shows how much attention should be given to the j^{th} word of the document when trying to predict the i^{th} label. Finally, the attention weight matrix $\mathbf{A}$ is multiplied with the hidden state matrix $\mathbf{H}$ to produce the label-specific vectors representing the input document as $\mathbf{V} = \mathbf{H A}^\top$, where each column $\mathbf{v_i}$ of the matrix $\mathbf{V} \in \mathbb{R}^{2u \times |L|}$ is the representation of the input document corresponding to label $\mathbf{c_i}$.

3.4.4 Output Layer

Each label-specific representation $\mathbf{v_i}$ is passed as input to a corresponding single-layer feed-forward network $\mathbf{FF_i}$ to produce the probability $\hat{\mathbf{y}}_\mathbf{i}$ of observing the i^{th} label given the document:

$$\hat{\mathbf{y}}_\mathbf{i} = \sigma(\mathbf{FF_i}(\mathbf{v_i})) \quad (11)$$

The training objective is to minimize the binary cross-entropy loss between the predicted label $\hat{\mathbf{y}}_\mathbf{i}$ and the target $\mathbf{y_i}$ where σ represents the sigmoid function.

4 Experiments

Following the state of the art work on ICD coding from clinical text, we test our model on the Medical Information Mart for Intensive Care (MIMIC) MIMIC-III (Johnson et al., 2016) dataset.

MIMIC-III Following previous work (Mullenbach et al., 2018), (Xie et al., 2019), (Li and Yu, 2020), we focus on the discharge summaries, which condense all the information during a patient stay into a single document. Each admission was tagged

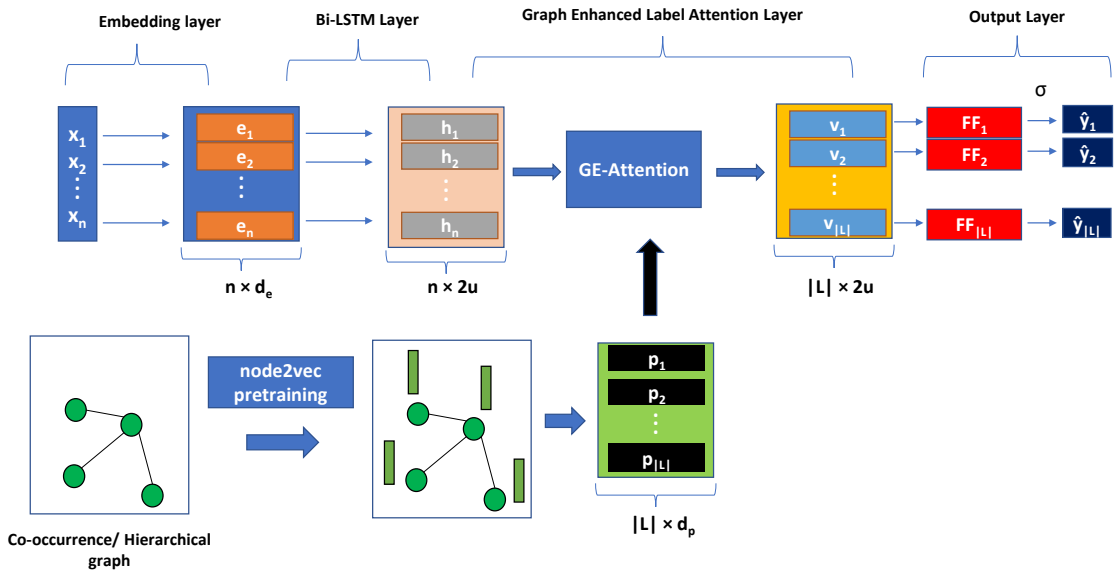

Figure 2: Network Architecture for GE-LAAT. The shapes are shown under the matrices. The input document vector, goes through the embedding layer and the Bi-LSTM layer, outputting the hidden state matrix. Then, by utilizing the pretrained node2vec vectors, the attention layer transforms the hidden state matrix into label-specific document vectors, which go through individual feed-forward neural networks.

manually by coders with a set of ICD-9 codes describing diagnoses and procedures during the patient stay. In this dataset, there were 52,722 discharge summaries and 8,929 unique codes in total. We use the same split provided by (Mullenbach et al., 2018) conduct experiments on the full set of codes. There are 47,719 discharge summaries for training, 1,631 for validation and 3,372 for testing. The exact preprocessing steps and the optimal hyperparameter settings in (Vu et al., 2020) are used in the model. For node2vec pretraining, we set $d_p = 128$ for GE-LAAT-C and $d_p = 512$ for GE-LAAT-H and use the default settings in the github repo [1] for the other parameters.

We present the results in Table 1. GE-LAAT-C and GE-LAAT-H represent the cooccurence and the hierarchical graph based versions of GE-LAAT respectively. GE-LAAT-H improves the state of the art Macro-AUC and Micro-AUC by 1.8% and 0.2% respectively. The improvement in Macro-AUC is more significant than Micro-AUC and GE-LAAT-C and GE-LAAT-H both have a higher Macro-AUC scores compared to LAAT and JointLAAT. This indicates a more balanced performance across all labels and suggests the graphs are useful for improving the performance for the rare labels. Moreover, GE-LAAT-H has a much higher Macro-AUC score compared to GE-LAAT-C, suggesting that the hierarchical graph is more useful than the cooccurence graph.

Model	AUC		F1		P@8
	Macro	Micro	Macro	Micro	
CNN	80.6	96.9	4.2	41.9	58.1
BiGRU	82.2	97.1	3.8	41.7	58.5
CAML	89.5	98.6	8.8	53.9	70.9
DR-CAML	89.7	98.5	8.6	52.9	69.0
MSATT-KG	91.0	99.2	9.0	55.3	72.8
MultiResCNN	91.0	98.6	8.5	55.2	73.4
HyperCore	93.0	98.9	9.0	55.1	72.2
GatedCNN-NCI	92.2	98.9	9.2	56.3	73.6
LAAT	91.9	98.8	9.9	**57.5**	**73.8**
JointLAAT	92.1	98.8	**10.7**	**57.5**	73.5
GE-LAAT-C	92.3	98.8	10.2	56.8	73.0
GE-LAAT-H	**93.8**	**99.0**	9.3	56.0	72.5

Table 1: Results for MIMIC-III full dataset

5 Conclusion

In this paper, we proposed a novel extension of LAAT by introducing a graph enhanced label attention mechanism. Our solution can learn useful code representations, which are then used to generate label-specific document vectors. Experiments show the effectiveness of our method.

[1] https://github.com/aditya-grover/node2vec

References

Anand Avati, Kenneth Jung, Stephanie Harman, Lance Downing, Andrew Y. Ng, and Nigam H. Shah. 2017. Improving palliative care with deep learning. *CoRR*, abs/1711.06402.

Elena Birman-Deych, Amy D Waterman, Yan Yan, David S Nilasena, Martha J Radford, and Brian F Gage. 2005. Accuracy of icd-9-cm codes for identifying cardiovascular and stroke risk factors. *Medical care*, 43(5):480—485.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. HyperCore: Hyperbolic and co-graph representation for automatic ICD coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114, Online. Association for Computational Linguistics.

Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Y. Guo. 2019. Multi-label image recognition with graph convolutional networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5172–5181.

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. Doctor ai: Predicting clinical events via recurrent neural networks.

Joshua C. Denny, Marylyn D. Ritchie, Melissa A. Basford, Jill M. Pulley, Lisa Bastarache, Kristin Brown-Gentry, Deede Wang, Dan R. Masys, Dan M. Roden, and Dana C. Crawford. 2010. PheWAS: demonstrating the feasibility of a phenome-wide scan to discover gene–disease associations. *Bioinformatics*, 26(9):1205–1210.

Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 855–864, New York, NY, USA. Association for Computing Machinery.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Shaoxiong Ji, Shirui Pan, and Pekka Marttinen. 2020. Medical code assignment with gated convolution and note-code interaction. *CoRR*, abs/2010.06975.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li wei H. Lehman, Mengling Feng, Mohammad Mahdi Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17.

Fei Li and Hong Yu. 2020. Icd coding from clinical text using multi-filter residual convolutional neural network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8180–8187.

J. Mullenbach, Sarah Wiegreffe, J. Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *NAACL*.

Rajesh Ranganath, Adler J. Perotte, Noémie Elhadad, and David M. Blei. 2015. The survival filter: Joint survival analysis with a latent time series. In *UAI*.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for icd coding from clinical text. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*.

Xiancheng Xie, Yun Xiong, Philip S. Yu, and Yangyong Zhu. 2019. Ehr coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 649–658, New York, NY, USA. Association for Computing Machinery.