공학석사 학위논문

Graph Convolutional Networks for Predictive Healthcare using Clinical Notes

헬스케어 예측을 위한 전자 건강 기록 기반 그래프 컨볼루션 모델

2020 년 7 월

서울대학교 대학원

컴퓨터 공학부

박 은 화

Graph Convolutional Networks for Predictive Healthcare using Clinical Notes

헬스케어 예측을 위한 전자 건강 기록 기반 그래프 컨볼루션 모델

지도교수 김 선

이 논문을 공학석사 학위논문으로 제출함

2020 년 6 월

서울대학교 대학원

컴퓨터 공학부

박 은 화

박은화의 공학석사 학위논문을 인준함

2020 년 6 월

위 원 장 장병탁

부위원장 김 선

위 원 송현오

Abstract

Graph Convolutional Networks for Predictive Healthcare using Clinical Notes

PIAO YINHUA
Department of Computer Science & Engineering
College of Engineering
Seoul National University

Clinical notes in Electronic Health Record(EHR) system are recorded in free text forms with different styles and abbreviations of personal preference. Thus, it is very difficult to extract clinically meaningful information from EHR clinical notes. There are many computational methods developed for tasks such as medical text normalization, medical entity extraction and patient-level prediction tasks. Existing methods for the patient-level prediction task focus on capturing the contextual or sequential information from clinical texts, but they are not designed to capture global and non-consecutive information in the clinical texts. Recently, graph convolutional neural networks(GCNs) are successfully used for text-based classification since GCN can extract the global and long-distance information among the whole texts. However, application of

GCN for mining clinical notes is yet to be fully explored.

In this study, we propose an end-to-end framework for the analysis of clinical notes using graph neural network-based techniques to predict whether a patient is with MRSA (Methicillin-Resistant Staphylococcus Aureus) positive infection or negative infection. For this MRSA infection prediction, it is critical to capture the patient-specific and global non-consecutive information from patient clinical notes. The clinical notes of a patient are processed to construct a patient-level graph, and each patient-level graph is fed into the GCN-based framework for graph-level supervised learning.

The proposed framework consists of graph convolutional network layer, a graph pooling layer and a readout layer, followed by a fully connected layer. We tested various settings of the GCN-based framework with various combinations of graph convolution operations and graph pooling methods and we evaluated the performance of each variant framework. In experiments with MRSA infection data, all of the variant frameworks with graph structure information outperformed several baseline methods without using graph structure information with a margin of 2.93%∼11.81%. We also investigated into graphs in the pooling step to conduct interpretable analysis in population-based statistical aspect and patient-specific aspect, respectively. With this inspection, we found long distance word pairs that are distinct for MRSA positive patients and we also showed the pooled graph of the patient that contributes to the patient-specific prediction. Moreover, the Adaboost algorithm was used to improve the performance further. As a result, the framework proposed in this paper reached the highest performance of 85.70%, which is higher than the baseline methods with a margin of 3.71%∼12.59%.

Keywords: Clinical notes, Graph Neural Network, Graph Pooling, Interpretable Analysis

Student Number: 2018-27910

Contents

List of Figures

List of Tables

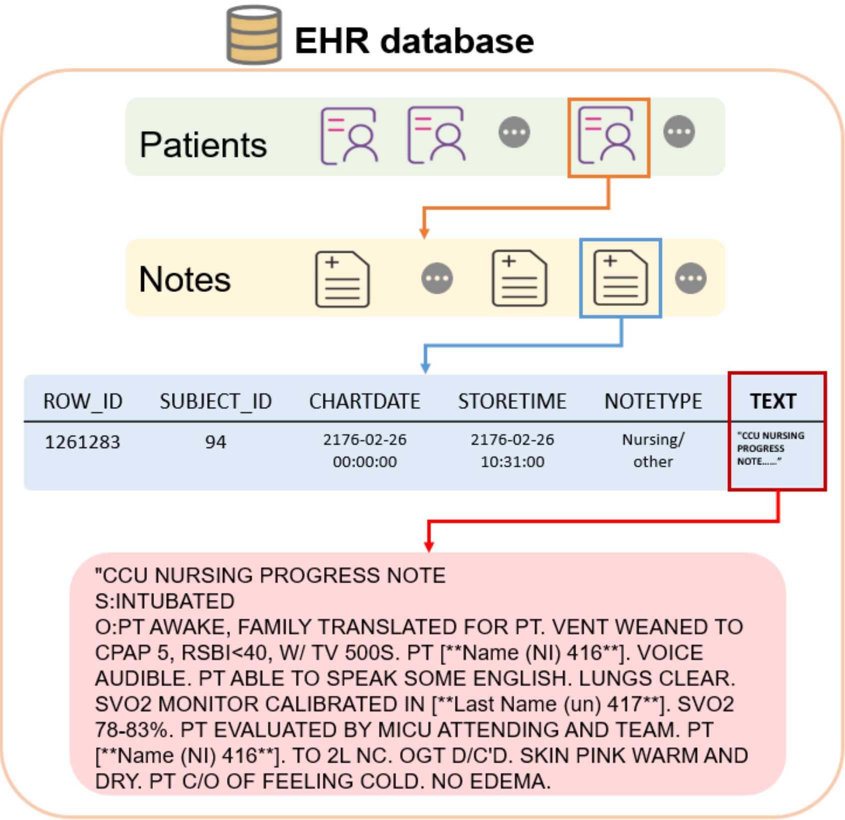
Chapter 1

Introduction

1.1 Background

1.1.1 EHR Clinical Text Data

Electronic Health Record(EHR) is digital version of a patient's paper chart. EHRs are real-time, patient-centered records that make information available instantly and securely to authorized users (`https://www.healthit.gov/faq/what-electronic-health-record-ehr`). EHR contains various data and large amount of patient information, such as demographic results, medical history, medication and allergy history, immune status, experimental test results and other information. There are generally two types of data in the EHR system. One is the structured data that is neatly stored in the form of medical codes, among which the most representative codes are diagnostic code (ICD-10) and lab result code(LOINC), etc. The other type is the unstructured data that refers to the patients' entire process in the hospital that contains rich information recorded in free text-based form. The computational framework proposed in this thesis is for the analysis of the unstructured patient clinical



ROW_ID	SUBJECT_ID	CHARTDATE	STORETIME	NOTETYPE	TEXT
1261283	94	2176-02-26 00:00:00	2176-02-26 10:31:00	Nursing/ other	"CCU NURSING PROGRESS NOTE……"

"CCU NURSING PROGRESS NOTE
S:INTUBATED
O:PT AWAKE, FAMILY TRANSLATED FOR PT. VENT WEANED TO CPAP 5, RSBI<40, W/ TV 500S. PT [**Name (NI) 416**]. VOICE AUDIBLE. PT ABLE TO SPEAK SOME ENGLISH. LUNGS CLEAR. SVO2 MONITOR CALIBRATED IN [**Last Name (un) 417**]. SVO2 78-83%. PT EVALUATED BY MICU ATTENDING AND TEAM. PT [**Name (NI) 416**]. TO 2L NC. OGT D/C'D. SKIN PINK WARM AND DRY. PT C/O OF FEELING COLD. NO EDEMA.

Figure 1.1: Patient Clinical Notes in EHR System

notes.

Clinical notes in EHR contains sequential information with the time stamps for patients. As depicted in the Figure 1.1, a patient can be represented by several clinical notes with irregular time-order information, and each clinical note can be represented by several sentences that are recorded by the medical staffs. There are also abundant types in the patient clinical notes which makes the clinical notes rich in author- and domain-specific idiosyncrasies. For example, nursing notes are recorded by nurses, ECG notes are recorded by real-time machines automatically, physician notes are recorded by the different doctors. Based on all of the aforementioned characteristics of the patient clinical notes, it is challenging to gain the insight from the patient clinical notes.

1.1.2 Current methods and limitations

With the advent of the era of big data and artificial intelligence, there are many computational methods in the field of natural language process(NLP) are developed for many tasks on the clinical texts, such as medical text normalization, medical entity extraction and patient-level prediction tasks. In the patient-level prediction task, the representation learning in patient-level becomes the most principal part. Existing methods learn the information from the patient clinical notes in different aspects.

Original methods used the bag-of-words model to simply represent the patient with the frequency of words, which could make insufficient use of the rich information and result in the curse of dimensionality. With the popularity of word embedding methods arising, some methods used the contextual information of the clinical texts to capture the semantic information by calculating the similarity of words and the patient representations can be mapped into a lower dimensional space. However, the contextual information from texts lacks of the time series information from the clinical notes.

Therefore, the sequential deep learning-based methods that are widely applied in the speech field are used for patient prediction by learning the time series information from the input sequence data in clinical notes. Since the patient data in EHR is represented hierarchically , most sequential models are either based on word-level that only consider the word-order information in single notes, or based on note-level that only consider the time-order information among the notes. Moreover, many experiments in previous works showed that the sequential-based model such as recurrent neural networks(RNNs) can only capture time series information limited to a short duration. Therefore, the sequential model can capture the local continuous semantic information from the clinical texts.

The graph-based method, which has recently emerged in text mining, uses

Graph Convolutional Neural networks(GCNs) to capture the graphical information in heterogeneous graph constructed by documents where the word-word nodes are connected by point-wise mutual information(PMI) and the word-document nodes are connected by TF-IDF values. The semi-supervised learning is conducted for text-based classification.

Likewise, the unstructured clinical text data can also be transformed into a structured graph by calculating the word co-occurrences. Using the constructed graph, GCN can capture global and long distance information from the clinical texts. In addition, as the relationships between nodes are preserved in edges, current GCN-based methods can not only predict interpretable results but also can map the patient into dense and informative embedding space where the global graph structure is preserved in patterns of connectivity.

1.2 Problem Statement and Contributions

Considering the hierarchical structure in patient-level representation and the diverse information in the patient clinical notes, we propose a framework based on graph convolutional neural networks to do the patient-level prediction task by extracting the global and non-consecutive information from the patient clinical notes. Furthermore, in order to retain the personalized characteristic of patient data in EHR, we construct graph for each patient to explore the individual-specific global information that contributes to the prediction.

INPUT The input of the proposed framework is the patient electronic health record consisting of several clinical notes.
OUTPUT The output of the proposed framework is the patient label. In this study, we predict whether the patient is infected with MRSA or not. The patient with MRSA positive result is represented by 1 and the one with MRSA negative result is represented by 0.

CONTRIBUTIONS

1) We proposed a graph-level graph convolutional neural network model for predictive healthcare on clinical notes using graph neural network-based techniques, which can capture patient-specific and global non-consecutive information.

2) In the graph pooling layer, we extract patterns from graphs that contribute to the predict results and conducting the interpretable analysis in population-based statistical aspect and patient-specific aspect, respectively.

3) Comparing with baseline methods, our framework using the global non-consecutive information outperforms other baseline models using the sequential or other form of information from clinical texts.

4) Applying the Adaboost algorithm to the framework for the better performance.

Chapter 2

Related Works

In this section, we introduce the state-of-the-art methods in text-based classification tasks and related works on patient classification using clinical notes.

2.1 Traditional Methods

Previous works on text-based classification are mainly focused on feature extraction. There are two taxonomies in extracting the features from the texts. One is hand-crafted feature engineering, which can use rule-based strategy to construct text representations in simple way. Bag-of-words is the representative method of hand-crafted feature engineering that represents a text by the discrete words without order information. Some commonly used linear classification methods are utilized for text classification using bag-of-words representations(Poulin *et al.*, 2014; Joffe *et al.*, 2015; Byrd *et al.*, 2014).

The other taxonomy in extracting the features from texts is automatic feature engineering, which can use simple model-based strategy to construct text representation automatically. Word2Vec (Mikolov *et al.*, 2013) leverages the contextual information from the texts to map sparse and high dimensional word vectors to a denser and lower dimensional vector space where the similar words are mapped to the similar positions. Also the informative embedding can be fed into the linear classification methods to do the downstream tasks. Choi *et al.* (2016a), Choi *et al.* (2016b) use the skip-gram methods that is one of the models in Word2Vec to encode medical codes.

2.2 Deep Learning Methods

Since the amount of clinical data is increased day by day as well as the popularity of the deep learning methods is risen recent years, more and more research focus on the deep learning methods to apply on the clinical fields.

The most representative methods of the deep learning methods are Convolutional Neural Networks(CNNs) and Recurrent Neural Networks(RNNs). Recurrent neural networks are proposed mainly for solving the time-series data. The temporal information of time-series data can be transformed from previous layer to the current layer by the gate units. Choi *et al.* (2017) first proposed the sequential model based on RNNs to predict heart failure patients using EHR medical codes, where the representation at each time point is combined with the time stamp and fed into the model. For the clinical notes, Sen *et al.* (2019) lists the different research based on RNNs using clinical notes, and these methods can be separated to two categories to learn the sequential representation. One is the note-level representation learning which regards each note as an input at a time point capturing temporal dependencies between the clinical notes, such as Dubois *et al.* (2017). To learn the representations in patient-level, Sen *et al.* proposed HAC-RNN utilizing the sequential information in both word and note levels, as well as hierarchical external attributes.

2.3 Graph Neural Networks

Inspired by the success of convolutional neural network(CNNs)(LeCun *et al.*, 1995) that leverages the properties of data such as images, speech, and video on Euclidean domains(grid structure), the convolutional and pooling operations in CNNs are redefined to apply on the graph data, such as social networks, biological networks, to exploit the property and characteristic of the data in non-euclidean domains.

2.3.1 Graph Convolutional Networks

As the combination of graph convolution operation and the neural networks is more efficient and convenient, the popularity of graph convolutional networks have grown rapidly in recent years. Graph convolutional network(GCN) is divided into two categories, spectral-based GCN (Bruna *et al.*, 2013; Defferrard *et al.*, 2016; Levie *et al.*, 2018) and spatial-based GCN (Kipf and Welling, 2016; Gilmer *et al.*, 2017; Hamilton *et al.*, 2017; Veličković *et al.*, 2017; Monti *et al.*, 2017; Xu *et al.*, 2018). The spectral-based GCN method introduces a convolutional filter from the perspective of graph signal processing to define the graph convolution operation that is interpreted as removing noise from the graph signal (Shuman *et al.*, 2013; Sandryhaila and Moura, 2013; Chen *et al.*, 2015).

Similar to the convolution operation of a conventional CNN on an image, spacial-based method is to define graph convolution operation based on the spatial relationship of the nodes. In general, the central node representation and the neighboring node representations are aggregated to update the new representation of the central node in order to combine more graph information. Unlike spectral-based GCN that takes a lot of time to calculate eigenvalue and Laplacian matrix, spatial-based GCN is very simple and produces the latest technological achievements in the graph classification tasks recently.The steps of spatial-based graph convolution is defined as follows:

$$h_v^k = COMBINE^k(h_v^{k-1}, AGGREGATE^k(h_u^{k-1} : u \in N(v))) \tag{2.1}$$

where the feature h_v^k of node v in the k-th iteration, depends on the *AGGREGATE*$^k(\cdot)$ function that aggregates the feature of neighboring nodes $h_u^{k-1}(u \in N(v))$ of node h_v^{k-1} and the $COMBINE^k(\cdot)$ function that combines neighboring nodes and the own features to update new features. Different spatial-based methods propose the different forms of $AGGREGATE^k(\cdot)$ function and $COMBINE^k(\cdot)$ function, such as Hamilton *et al.* (2017), Veličković *et al.* (2017), Xu *et al.* (2018) and so on.

2.3.2 Graph Pooling Methods

In order to construct a model of the computational capability and the interpretability of the graph network, it is essential to downsample the graph and reduce the graph size. There are fewer graph pooling methods than graph convolution methods. Previous work (Dhillon *et al.*, 2007) used topology-based graph coarsening algorithm through the spectral clustering, which coarsens the graph by the feature decomposition resulting in the high computational complexity .

Global pooling methods do simple operation, such as add, mean, max process or neural network, to pool all the nodes into one dimensional vector space. Since all of the node representations are pooled at once in global pooling methods. In addition, Set2Set(Vinyals *et al.*, 2015) and Sortpool (Zhang *et al.*, 2018) use a bit more complex methods, such as attention mechanism and ranking the scores, to conduct the global pooling process.

Inspired by the pooling methods in CNNs, the hierarchical pooling methods are transferred into the graph pooling methods in which the node representations can be learned layer-by-layer using both node features and graph structures. It is important to capture hierarchical information of the graph structure in the pooling step. Diffpool(Ying *et al.*, 2018) is an end-to-end method that the soft assignment node matrix calculated by GCN is leveraged to divide fixed-size node clusters, and then each node in the pooled graph is represented by aggregating all nodes in each cluster. However, the computational complexity reaches $O(n^2)$. Recently, SAGPool proposed by Lee *et al.* (2019). not only considers the node features and graph structures but also greatly reduces the computational complexity using the self-attention mechanism(Vaswani *et al.*, 2017).

2.3.3 Applications of GNN

The applications of the graph neural networks has resulted in outstanding performance in the various fields, such as recommendation systems(Berg *et al.*, 2017; Yao *et al.*, 2018; Monti *et al.*, 2017), chemical researches(You *et al.*, 2018; Zitnik *et al.*, 2018), natural language processing(Bastings *et al.*, 2017; Yao *et al.*, 2019; Peng *et al.*, 2018).

In the natural language processing, TextGCN proposed in Yao *et al.* (2019) constructed the heterogeneous graph to classify the documents by capturing global mutual information of the documents , where the edge weight between word node and word node are calculated by co-occurrence positive point-wise mutual information and the edge weight between word node and document node is represented by TF-IDF. The framework of training graph convolutional neural network is constructed by the semi-supervised learning for the document node classification. However this framework is not satisfied to the patient graph data with high idiosyncrasy from patient to patient.

Recently Choi *et al.* (2019) proposed a framework to learn the graphical structure from EHR data with graph convolutional transformer, which leverages prior knowledge and graph convolutional networks to initialize the first step and directs the transformer to do the link prediction between two medical codes. As can be seen, graph neural network models have not been applied to the clinical texts, which is not only to transform the text contents to the graph but capturing global and patient-specific information from the graph.

Chapter 3

Methods and Materials

3.1 Notation and Problem Definition

Throughout this paper, we use uppercase characters to denote matrices and lowercase characters to denote vectors. We represent the set$\{1, \cdots, n\}$ by $[n]$ in the rest of the paper. Unless particularly specified, the notations used in this paper are illustrated in Table 3.1. Now we define the minimal set of definitions required to understand this paper.

Definition 1.(*Clinical Notes*) Clinical notes for each patient are represented as a sequence of the text notes. The t-th patient $P^{(t)}$ is represented by a sequence of $|d^{(t)}|$ clinical notes. For patient $P^{(t)}$, each note $d_i^{(t)}$ contains $|d_i^{(t)}|$ words, in which $w_{ij}^{(t)}$ denotes the j-th word in i-th note from t-th patient, where $j \in [|d_i^{(t)}|]$ and $i \in [|d^{(t)}|]$.

Definition 2.(*Graph*) A graph is represented as $G = \{V, E\}$ where $V(|V| = n)$ is the set of vertices or nodes(we use nodes throughout the paper), and E is the set of edges. Let $v_i \in V$ to denote a node and $e_{ij} = (v_i, v_j) \in E$ denote an edge from v_i to v_j. Let $X \in \mathbb{R}^{n \times m}$ be a matrix containing all n nodes

Table 3.1: Commonly Used Notations

Notations	Descriptions
$P^{(t)}$	t-th patient in EHR dataset.
$d_i^{(t)}$	i-th note in t-th patient.
$w_{ij}^{(t)}$	j-th word in i-th note from t-th patient.
$\lvert P^{(t)}\rvert$	the number of words in t-th patient.
$\lvert d^{(t)}\rvert$	the number of notes in t-th patient.
$\lvert d_i^{(t)}\rvert$	the number of words in i-th note from t-th patient.
$N(v)$	the neighbors of a node v.
n	the number of nodes, $n = \lvert V\rvert$.
m	the dimension of a node feature vector.
x_v	the feature vector of the node v.
$h_v \in \mathbb{R}^d$	the hidden feature vector of node v.
$h_G \in \mathbb{R}^{2d}$	the hidden feature vector of graph G.
$X \in \mathbb{R}^{n\times m}$	the node feature matrix of a graph.
$X^{(k)}, H \in \mathbb{R}^{n\times d}$	the node feature matrix in k-th graph convolutional layer.
$H_{out} \in \mathbb{R}^{\lceil rn\rceil\times d}$	the node feature matrix in pooling layer.
$W, P, Q, \Theta, \theta, \epsilon$	Learnable model parameters.

with their features, where m is the dimension of the feature vectors, each row $x_v \in \mathbb{R}^m$ is the feature vector of v. The adjacency matrix A is a matrix with $A_{ij} = 1$ if $e_{ij} \in E$ and $A_{ij} = 0$ if $e_{ij} \notin E$.

Definition 3.(*Patient Graph*) A patient graph is represented as $G^{(t)} = \{V^{(t)}, E^{(t)}\}$ where $V^{(t)}(|V^{(t)}| = n)$ is the set of words from the records of patient $P^{(t)}$, and $E^{(t)}$ is the set of co-occurrences of these words from the records of the patient, and the patient graph is undirected and unweighted. If two nodes are connected, there is a pair of edge with inverse in the undirected graph

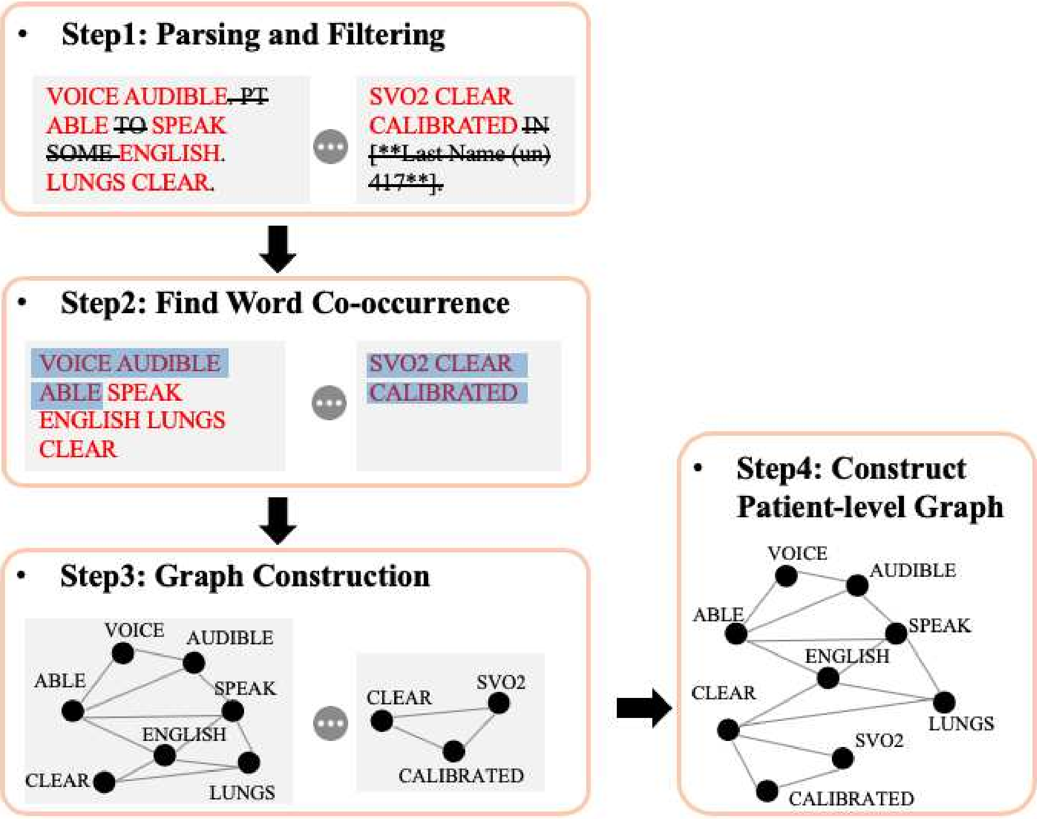


Figure 3.1: Patient Graph Construction Process

where the adjacency matrix is symmetric.

The goal is to learn a model that predicts the label $y^{(t)}$ belongs to $\{0, 1\}$ for a new patient graph given the clinical note set for the patient. True label $y^{(t)}$ indicate the positive or negative case of infection. Since we focus on a single patient in this study, we omit the superscript (t) throughout the paper.

3.2 Patient Graph Construction Process

In order to leverage graphical structure from the patient clinical notes, a patient representation can be viewed as a patient graph. In this section, we introduce how to construct the patient graph that contains non-consecutive and long distance semantic information by using patient clinical notes. As shown in Figure 3.1, there are 4 steps in constructing the patient graph. First, parsing and filtering the raw text of patient clinical records and finding the word co-occurrences for each clinical note. After constructing note-level graphs using the word co-occurrences for the patient, we combined all these note-level graphs to construct a patient-level graph to represent the patient.

3.2.1 Parsing and Filtering

In order to learn more effective structural information from the graphs and achieve better performance in predicting the patient labels, the words from all notes are filtered by fold change value in the whole data set. Formally, We calculate the global frequency of each word both from positive samples and negative samples. The words are ranked by fold change value following the formulation:

$$FC(w_k) = \frac{\frac{\#C_{pos}(w_k)}{\#C_{pos}}}{\frac{\#C_{neg}(w_k)}{\#C_{neg}}}, \tag{3.1}$$

where C_{pos} and C_{neg} denote the total number of windows sliding on the positive and negative samples, respectively. $C_{pos}(w_k)$ and $C_{neg}(w_k)$ represent the number of windows in positive and negative samples where the word w_k occurred. Top 30% and bottom 30% words are filtered from the global word set in the whole samples, which means that the filtered words can better distinguish the positive and negative samples than other words. In this way, the word set from clinical notes is filtered into a smaller size in order to a better performance of prediction.

3.2.2 Word Co-occurrence Finding

We first construct graph for each clinical note. In order to construct graph using word co-occurrences, we apply the sliding window algorithm that is commonly used in the text-to-graph transformation. A fixed-size sliding window is pre-defined and used for sliding on the clinical note. As shown in the second step in Figure 3.1, for example, [“VOICE”, “AUDIBLE”], [“AUDIBLE”, “ABLE”] and [“VOICE”, “ABLE”] cooccurred in the 3-size window from the first note. Regarding each word, that is remained in the previous filtering step, as a node in the graph, we construct an edge between two nodes if they co-occur in the current window. Using these co-occurrences of words, a graph can be constructed in note-level, which can be represented by:

$$G_i = \{V_i, A_i\}, \tag{3.2}$$

where $i \in [|d|]$, and the clinical note d_i can be viewed as a graph G_i consisting of $|d_i|$ of nodes with an adjacency matrix A_i that describes the connections between the nodes from the clinical note d_i.

3.2.3 Patient-level Graph Representation

The motivation of this paper is to capture patient-specific and non-consecutive information from the patient clinical notes. Moreover, the graph construction is already known to connect long distance information. Therefore, aggregating all of these note-level graphs constructed in the previous step for the patient can not only extract global non-consecutive information among the clinical notes from the patient but can learn the patient-specific characteristic feature by training samples one by one. Thus, we aggregate the note-level graphs to a patient-level graph that is undirected and unweighted, following the formulation:

$$G = [G_1, ..., G_{|d|}] = \{V, A\}, \tag{3.3}$$

where the patient can be viewed as a graph G consisting of the set of words $V(|V| = |P|)$ with an adjacency matrix A that describes the connections between the nodes from patient P, which aggregates the all of the information from the clinical notes of the patient P into one patient-level graph G.

3.3 Word Embedding

Before introducing the graph neural networks for the patient clinical graph, as the input graph constructed in the previous section is composed of co-occurrences of words that are represented by real-valued vectors where semantically similar words are mapped close to each other. Therefore, we learn the word embedding using the skip-gram model of the 'word2vec'(Mikolov *et al.*, 2013), which trained prior to end-to-end training model separately.

As introduced in the related works chapter, the main progress of word2vec model is to map each raw representation(one-hot encoding) to a denser embedding space where implicitly contains the similarity among words by training a predictive machine learning model. Intuitively, given a target word w_i from a sentence $S = [w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}]$, skip-gram method regards the target word w_i as the input to predict the contextual words $(w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2})$ of the target words w_i which is depicted in Figure 3.2.

In order to further accelerate the training, we apply the training tricks of hierarchical softmax and negative sampling. As a result, words are mapped to vectors using $x_i = \theta_{w_i}$ where $\theta_{w_i} \in \mathbb{R}^m$ are the learned embedding vectors of word w_i from the skip-gram model. And each patient graph can be represented by

$$G = \{X, A\}, \tag{3.4}$$

where $X \in \mathbb{R}^{n \times m}$ and $x_i \in X$.

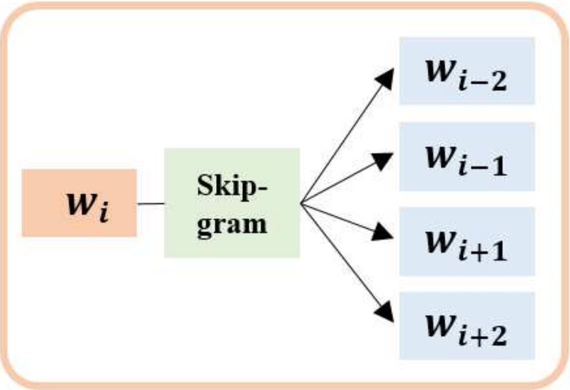


Figure 3.2: Word2Vec(skip-gram model)

3.4 Model Architecture

In this section, we introduce the model architecture that is mainly based on graph neural network. As depicted in Figure 3.3, given the patient graph G that is constructed in the previous section, the model proposed in this paper conducts supervised learning in an end-to-end fashion through the graph convolution layer, graph pooling layer, and readout layer:

1) The graph convolutional layer is used to embed the high-level representations of the nodes where the graph representation is transformed to G_{conv}.

2) The graph pooling layer acts as a downsampling function, which coarsens graph into a smaller graph G_{pool} with more essential information of the patients.

3) In order to implement the construction of the personalized model for patients, we perform graph-level prediction using the readout layer to aggregate the node representations in the graph into one dimensional embedding vector h_G to represent the graph.

Finally, multi-layer perceptrons and softmax layers are applied to graph-level representation and we can build an end-to-end framework for the graph

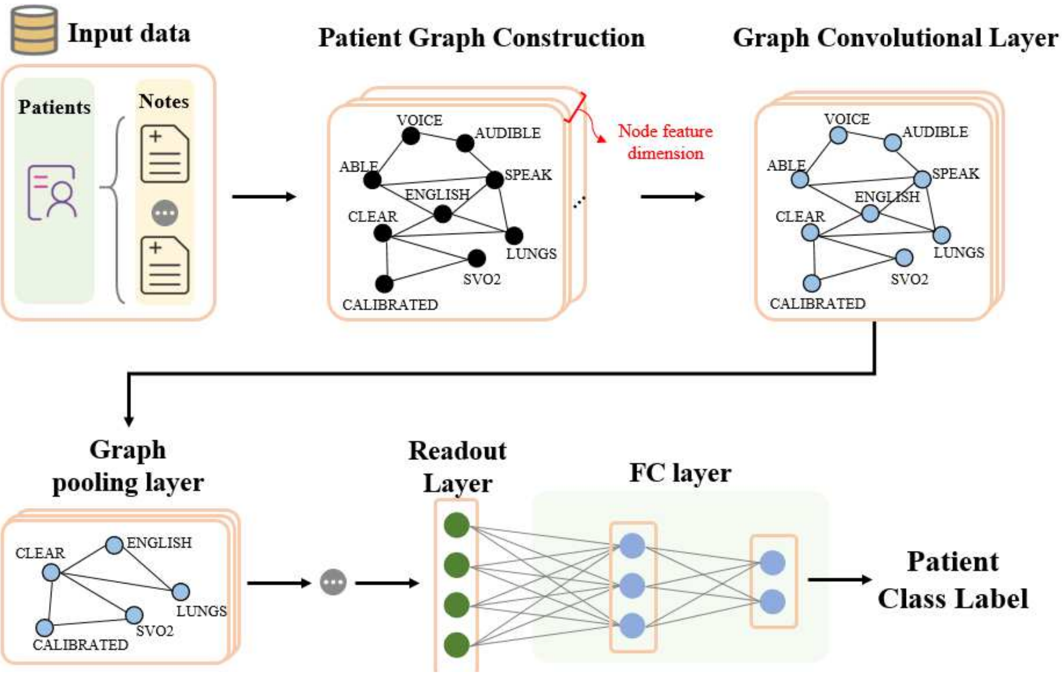


Figure 3.3: Model architecture

classification. Details of the methods that are used in each module are introduced in turn as below.

3.4.1 Graph Convolutional Network layer

We apply the graph convolutional networks that are mentioned in the related works to the patient graphs where the node representations can message each information to each other, aggregate information from each other and update the representations. we update the node representations and transform the graph G to a new representation G_{conv} by using vanilla GCN(Kipf and Welling, 2016) and GIN(Xu *et al.*, 2018), respectively.

Vanilla Graph Convolutional Network

The first methods is vanilla GCN that bridged the gap between spectral-based methods and spatial-based methods. Since it is proposed, spatial-based meth-

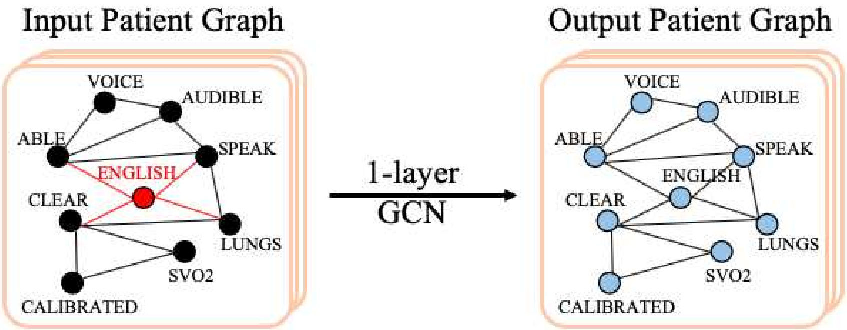


Figure 3.4: 1-layer spatial-based GCN. For example, the representation of node"ENGLISH" is aggregated and updated with the representations from its 1-hop neighbor nodes("ABLE", "SPEAK", "CLEAR", "LUNGS"). Output patient graph is the transformed graph with hidden node features after 1-layer GCN.

ods have rapidly developed due to their compelling efficiency and versatility. Vanilla GCN uses a first-order approximation to simplify calculations based on the convolution of spectral graphs (Levie *et al.*, 2018) for network models that directly operate on graph-structured data and proposes a simple and effective propagation method through layer-by-layer. For a one-layer GCN, the new m-dimensional node feature matrix $X^{(1)} \in \mathbb{R}^{n\times m}$ is computed as

$$X^{(1)} = \sigma(\widetilde{A}X^{(0)}W_0) \tag{3.5}$$

where $\widetilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix and $W_0 \in \mathbb{R}^{m\times d}$ is a weight matrix, d is the size of the convolution filter. σ is an activation function. A is adjacency matrix where the diagonal elements are set to 1 as every node is assumed to be connected to itself. Degree matrix D can be calculated from the graph where $D_{ii} = \sum_j A_{ij}$.

Through the first-order approximate simplification method, a single-layer vanilla GCN can be used to process information on the 1-hop neighbors in the graph as shown in the Figure 3.4. To deal with higher-order neighbors, a multi-layer vanilla GCN can be stacked to capture the localized feature of the graph network to build node representations from the k-hop neighborhood of each node:

$$X^{(k)} = \sigma(\widetilde{A}X^{(k-1)}W_{k-1}) \tag{3.6}$$

The graph representation G is transformed through the vanilla GCN layer as follow:

$$G_{conv} = GCN(G = \{V, A\}) = \{X^{(k)}, A\} \tag{3.7}$$

Graph Isomorphism Network

Spatial-based GCN methods are also called Massage Passing Neural Network(MPNN), which update node representations by recursively iterating and combining the first-order neighborhood representations to increase the size of receptive field. As shown in the related works, different MPNN-based methods propose different forms of $AGGREGATE^k(\cdot)$ and $COMBINE^k(\cdot)$.

However, in terms of classifying the graphs that have different graph structures, the embedding learned by previous MPNN-based methods proved to be incapable of effectively capturing the different information from the different graph structures. Besides, there are definitely differences among the structures of patient graphs that we constructed before, so we used the simple and powerful Graph Isomorphism Network(GIN) proposed by Xu et al., which proved that Weisfeiler-Lehman(WL) test(Douglas, 2011) is powerful and proved that if $AGGREGATE^k(\cdot)$, $COMBINE^k(\cdot)$ are injective functions, then GCN can be as powerful as WL test. GIN adjusts the weight of the central node by a learnable parameter ϵ^k and performs graph convolution by

$$h_v^k = MLP^k((1 + \epsilon^k) \cdot h_v^{k-1} + \sum_{u \in N(v)} h_u^{k-1}) \tag{3.8}$$

where $MLP(\cdot)$ represents a multi-layer perceptron combined with the process of summing up the neighboring node features can fit the formulation to an

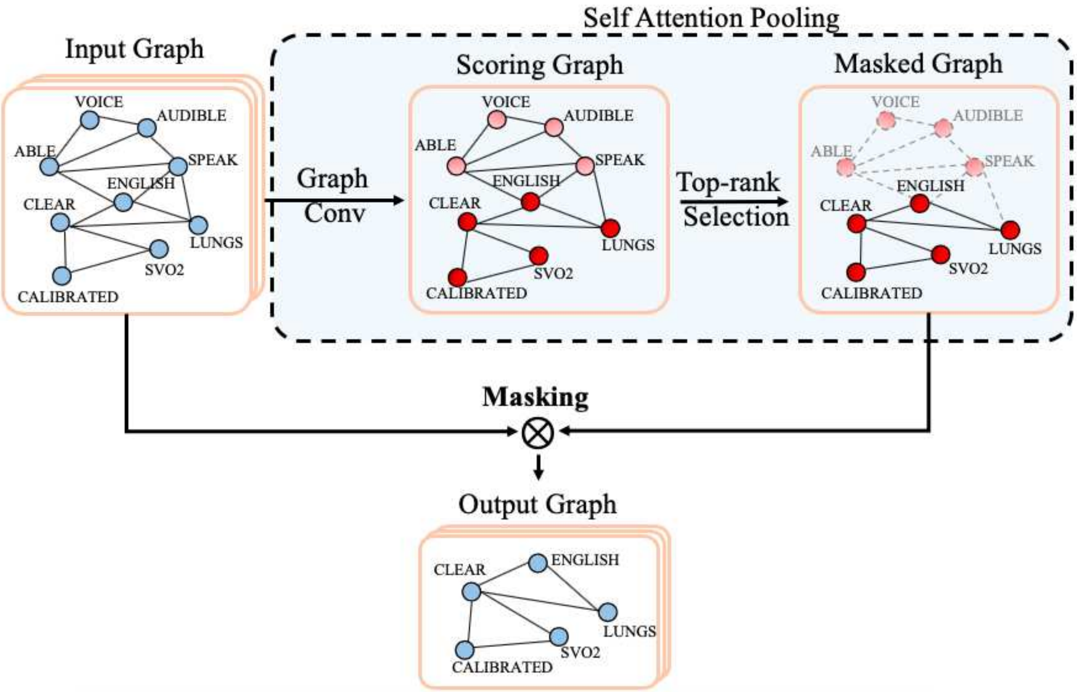


Figure 3.5: Self-Attention Graph Pooling

injective function.

The graph representation G is transformed through the GIN layer as follow:

$$G_{conv} = GIN(G = \{V, A\}) = \{X^{(k)}, A\}, \tag{3.9}$$

where $X^{(k)} \in \mathbb{R}^{n \times d}$ is the matrix of n node features $h_v^k \in \mathbb{R}^d$.

3.4.2 Graph Pooling layer

In order to reduce the graph size and find more significant patterns in the graph, we use self attention graph pooling proposed in Lee *et al.* (2019), which can use node features and topology to extract hierarchical representations with a reasonable complexity of time and space. As shown in the Figure 3.5, we first calculate self-attention score $Z \in \mathbb{R}^{n \times 1}$ using graph convolution operation. if we use the vanilla GCN , the node scoring matrix is calculated by

$$Z = \sigma(\widetilde{A} H \Theta_{att}) \tag{3.10}$$

where σ and $\widetilde{A}$ are the same meaning as mentioned in previous section. $H = X^{(k)} \in \mathbb{R}^{n\times d}$ is the output matrix in k-layer convolution operation with n nodes and d dimensional features, and $\Theta_{att} \in \mathbb{R}^{m\times 1}$ is the only parameter of the pooling layer.

The graph convolution operation in the SAGPool leverages the graph structure information to assign score to each node for pooling which takes not only node feature but also graph structure into consideration to coarsen the graph that preserves the graph structure information. We apply three different GCNs to the SAGPool model which is spectral-based GCN, vanilla GCN and GIN respectively and performance comparisons are shown in the Chapter 5.

According to the attention value Z to select nodes that have higher scores by adopting the node selection method of Gao and Ji (2019). If the pooling ratio $r \in (0, 1]$ is to determine the number of nodes to keep in next layer, the top $\lceil rn \rceil$ nodes are selected based on matrix Z by

$$idx = \text{top-rank}(Z, \lceil rn \rceil), Z_{mask} = Z_{idx} \tag{3.11}$$

where top-rank is the function that returns the indices of selected nodes and then indexing the selected nodes to the feature attention mask Z_{mask}. Finally, an input graph is pooled by the operation notated as masking as follow:

$$H' = H_{idx,:}, H_{out} = H' \odot Z_{mask}, A_{out} = A_{idx,idx} \tag{3.12}$$

where $H_{idx,:}$ is the node-wise indexed feature matrix, $\odot$ is the broadcasted element-wise product, and $A_{idx,idx}$ is the row-wise and col-wise indexed adjacency matrix. H_{out} and A_{out} are the new feature matrix and the corresponding adjacency matrix after pooling, respectively.

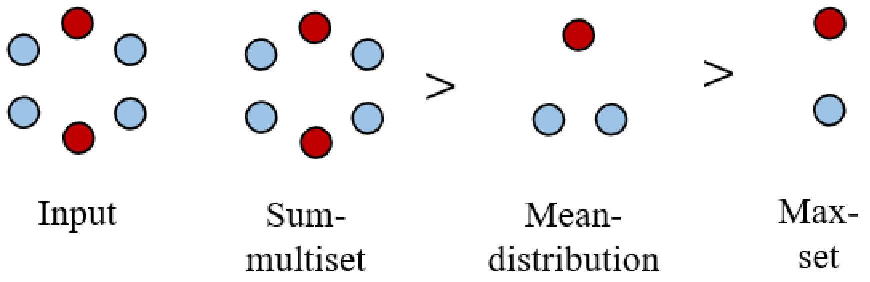


Figure 3.6: Comparisons about three aggregation methods in graph neural networks: Sum, Mean, Max from Xu *et al.* (2018).

The graph representation G_{conv} is coarsened through the SAGpooling layer as follow:

$$G_{pool} = SAGPool(G_{conv}) = \{H_{out}, A_{out}\}, \tag{3.13}$$

where $H_{out} \in \mathbb{R}^{\lceil rn \rceil \times d}$ is the matrix of $\lceil rn \rceil$ node features.

3.4.3 Readout Layer

Inspired by the theorem proposed in Xu *et al.* (2018) about graph-level readout functions, as shown in the Figure, the color of nodes represents the node feature and three aggregations are ranked by their representational power. The sum aggregation captures the most important information of graph structure than mean and max aggregation methods. The sum aggregation captures the full multi-set, however, the mean captures the proportion or distribution of elements of a given type and the max aggregation ignores multiplicities(reduces the multi-set to a simple set). Therefore we apply the concatenate the results using sum aggregation and mean aggregation, which can globally capture the structural information from graph and distribution information from node feature in the same time.

$$h_G = CONCAT(sum(h_v|v \in G_{pool})|max(h_v|v \in G_{pool})) \tag{3.14}$$

where $h_v \in \mathbb{R}^d$ denotes the hidden feature vector of node v after the pooling layer, and $h_G \in \mathbb{R}^{d\times 2}$ denotes the one dimensional embedding vector representing the graph G which can be used in the downstream task, such as classification, clustering, etc.

3.5 Prediction and Loss Function

The coarsened one-dimension embedding vector h_G can be seen as a dense embedding for patient graph G. Given the vector h_G of the graph G, the embedding vector h_G for graph G is then passed to MLPs, which outputs scores $Y_{pred} \in \mathbb{R}^C$ for each class:

$$Y_{pred} = P\mathrm{ReLU}(Qh_G), \tag{3.15}$$

where $P \in \mathbb{R}^{2d\times C}$, $Q \in \mathbb{R}^{2d\times 2d}$, C is the number of the patient classes. Finally, given the ground truth label Y and outputs scores Y_{pred}, we minimize the binary cross-entropy loss between the predicted labels and ground truth labels, shown as follow.

$$Loss = -(Ylog(y_{pred}) + (1-Y)log(1-y_{pred})) \tag{3.16}$$

3.6 Adaboost algorithm

We also apply the ensemble model to our model in order to capture more important information and improve the model performance. Adaboost(Hastie *et al.*, 2009) is an iterative algorithm, and the core idea of Adaboost is to train different classifiers(weak classifiers) against the same training set, and then combine these weak classifiers to form a stronger final classifier.

As shown in Figure, we view a GNN model that is introduced previously as a weak classifier. If there are k(k=3) weak models constructed in Adaboost, we calculate the error rate e_k for each weak model and using error rate e_k

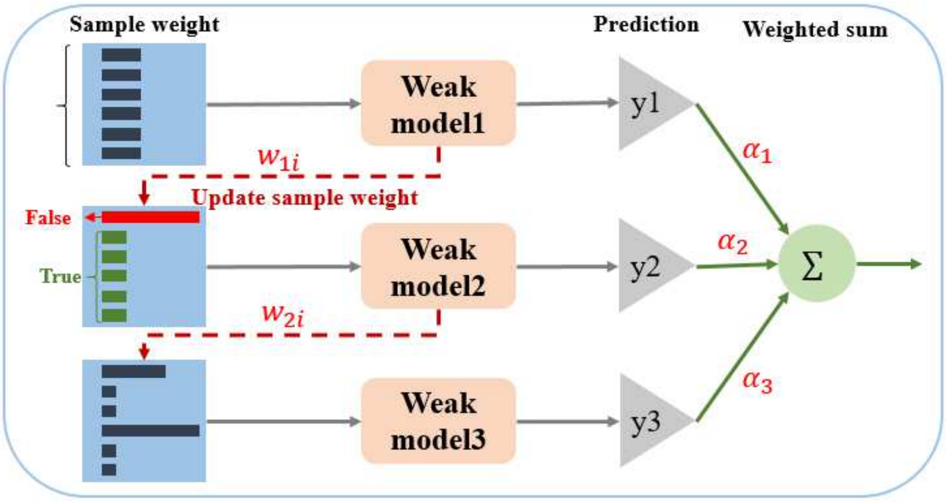


Figure 3.7: Overview of Adaboost Algorithm

calculate the weight α_k of k-th weak model:

$$e_k = P(M_k(p_i) \neq Y_i) = \sum_{i=1}^{N} w_{ki} I(M_k(p_i) \neq Y_i) \tag{3.17}$$

$$\alpha_k = \frac{1}{2} log \frac{1 - e_k}{e_k} \tag{3.18}$$

where M_k denotes the k-th weak model in the iteration, p_i denotes the input graph data of i-th patient and w_{ki} represents the patient p_i's weight in the k-th weak model.

In order to input the sampled the patient data to next weak model, the patient's weight w_{k+1}, i is updated by increasing the weight of incorrectly predicted patient and decreasing the weight of correctly predicted patients in the previous weak model M_k.

$$w_{k+1,i} = \frac{w_{k,i}}{Z_k} exp(-\alpha_k Y_i M_k(p_i)), \tag{3.19}$$

where Z_k is a factor for normalization. After several iterations, the results from each weak model are weighted summed to a final result of Adaboost.

$$f(p) = sign(\sum_{k=1}^{3} \alpha_k M_k(p)) \tag{3.20}$$

Chapter 4

Experiments

4.1 EHR Dataset

4.1.1 Introduction to MIMIC-III Dataset

In order to evaluate our framework, we use the real-world data from the publicly-available critical care database MIMIC III (Johnson *et al.*, 2016). MIMIC-III integrates de-identified, comprehensive clinical data of patients admitted to the Beth Israel Deaconess Medical Center in Boston, Massachusetts. It contains all unstructured clinical notes from caregivers of 45,000 patients. The resource of clinical notes is obtained from the table named 'NOTE-EVENTS' including nursing and physician notes, ECG reports, radiology reports and discharge summaries. We use notes from all categories except for discharge summaries that contains the information at the end of the hospital stay, which allows the trained model to predict in real-world applications.

4.1.2 MRSA Data Collection

Staphylococcus aureus is one of the most common causes of Hospital-Acquired Infections(HAIs), and Methicillin-Resistant Staphylococcus Aureus(MRSA) is one antibiotic-resistant strain of this bacteria. MRSA infections may result in serious complications including sepsis and death. Therefore we experiment with prediction problem using MRSA cohorts extracted from the MIMIC-III dataset.

To identify MRSA-positive patients, we extract the microbiology test associated with the organism **80293**(MRSA), found in the *Microbiology Events* table. Consequently, we extract all 1,228 patients who have a record of this test as our MRSA-positive patients. As the majority of patients in MIMIC dataset do not contract MRSA, we randomly subsample patients who have no record of a test for organism **80293** as MRSA-negative patients that has the same size as MRSA-positive patients.

Before training the model, we conduct the preprocessing for the raw MRSA data. We calculate the mean number of notes over all patients as the max length of notes to discard the first notes for who has more notes based on Hartvigsen *et al.* (2018) and Sen *et al.* (2017), which revealed that the symptoms should appear nearing discharge because MRSA is caused by bacteria with short incubation periods. In addition, patients whose graph size is less than 10 are filtered in order to reduce noise of the model performance. The statistical information for raw and preprocessed MRSA dataset are shown in Table 4.1, respectively.

4.2 Hyper Parameter Settings

The clinical notes are split into a set of words using NLTK package(Loper and Bird, 2002), and stop words are removed. Meanwhile, the words are trans-

Table 4.1: Statistic of MRSA Dataset

Statistic		MRSA(raw)	MRSA(preprocessed)
# Notes/Patient	Mean	16	16
	Median	20	20
# Unique Words/Patient	Mean	549	114
	Median	546	110
# Unique Words		32,256	10,336
# Patients		2,456	2,398

formed into lowercase and punctuation is removed from clinical notes. we use 5 fixed-length window sliding the clinical texts to find word co-occurrences. And the input node embedding is obtained by leveraging all notes from training set to pre-train the Word2Vec model where the skip window size is set to 1, number of skips to 2 and the number of negative examples to 64.

4.2.1 Model Training

To achieve the optimal prediction results, the hyper-parameters during training the model are set as follows: the dimension of word embedding is set to 32, the dimension of graph convolution filter is set to 64, the number of layers of graph convolution and graph pooling is set to 2, the number of MLP layers is set to 2, dropout probability is set to 0.5 and batch size is set to 32. We use the Adam optimizer with a learning rate decay strategy where an initial learning rate is set to 0.001 and the reduce factor is 5e-4 and the training is stopped when the loss of validation set does not decrease anymore. The size rate of training set, validation set and test set is split into 8:1:1 and we report mean and standard deviation of the results over 5 runs with 5 different random seeds.

4.3 Baseline Models

In general, We design the variant models based on aforementioned framework to show the performance of different combinations of graph convolution operations and graph pooling methods. Besides, we also implement state-of-the-art methods as baselines to compare with our framework.

- **Bag-Of-Word Average embedding + MLP**: We combine patient notes into one document. Using all unique words that filtered in the step of parsing and filtering (Table 4.1), a bag-of-words representation is created by the word frequency in patient notes. And we average the representations with normalization. These representations are then fed into MLPs that has the same construction as the MLPs in our framework.

- **Word2Vec + MLP**: We concatenate the word embedding that is learned in Word2Vec and use the sum aggregation and mean aggregation to represent a patient, which has the same process as the readout layer does in our framework. Also, the same construction of MLPs are then trained. This model will be the basis to evaluate whether the patient graph structure learned by our framework is useful at all.

- **Word2Vec + LSTM**: We concatenate the learned word embedding in the same way to represent each note of a patient. For each patient, we set the max length of notes to 20 and pad the sequences up to 20 with vectors of 0's for the patients whose note length is fewer than 20. These sequences consisting of note-level representations are then fed into a LSTM for final prediction.

- **GCN w/o pooling**: From our framework, we combine two-layer vanilla GCNs and readout layer to obtain the patient graph representations and fed them into MLPs to do the prediction.

- **GIN w/o pooling**: From our framework, we combine two-layer GINs and readout layer to obtain the patient graph representations and fed them into MLPs to do the prediction.

- **GIN w/ SAGPool**: We implement the hierarchical pooling architecture from the recent hierarchical pooling study. There are two blocks each of which consist of GIN and graph pooling methods based on Self Attention Pooling. We conduct different variant pooling methods based on SAGPool where the scoring matrix is calculated by spectral-based GCN method and spatial-based GCN methods as follows:

  - **Pool(GraphConv)(default)**: GraphConv (Morris *et al.*, 2019) is spectral-based and default method that is leveraged to calculate scoring matrix in proposed SAG Pooling method.
  - **Pool(Vanilla GCN)**: We replace spectral-based GraphConv with Vanilla GCN to calculate scoring matrix in SAG Pooling.
  - **Pool(GIN)**: GIN is leveraged to calculate scoring matrix in SAG Pooling.

  The outputs of readout layer are fed into the MLPs for classification.

Chapter 5

Results

5.1 Performance Comparisons with baseline models

MLP with pre-trained word embedding outperforms the one with bag-of-words word embedding which reveals that pre-trained embedding consisting of word similarity information is more important than bag-of-words embedding that contains discrete and simple frequency information.

LSTM showed worst performance among the baseline models and the reason seems to be that the sequential model is sensitive for input data with irregular-length that needs to be padded in a fix-length sequence before being fed into the model, which will lose a part of information in representing the note of a patient. Therefore it is proved that sequential information can not be captured easily from the patient clinical text data.

Since the architecture used in MLP is the same as the one used in graph-based methods, the only one difference between MLP and graph-based methods is that graph-based methods leverage the patient graph structure information for prediction and MLP predicts patient label without graph structure in-

Table 5.1: Performance comparison of baseline methods and graph-based methods(Test-set AUC). (Input embedding is pre-trained by Word2Vec except for “MLP+BOW”.)

Type	Methods	AUC(mean±std)
Baselines	MLP+BOW	0.7982±0.0165
	MLP	0.8145±0.0190
	LSTM	0.7475±0.0300
GCNs	Vanilla GCN	0.8157±0.0304
	GIN	0.8327±0.0161
GIN w/ SAGPool	Pool(default)	0.8258±0.0330
	Pool(Vanilla GCN)	0.8255±0.0222
	Pool(GIN)	**0.8492±0.0136**

formation. As the performance comparison between the MLP and any method of graph-based methods shown in the results of ‘GCNs’ in the Table 5.1, the graph structure information learned in graph-based methods does contribute to the patient prediction.

5.2 Performance Comparisons with graph networks

In general, the graph-based methods outperform baseline methods which seems to indicate that the global non-consecutive and patient-specific information from constructed patient graph in our work is important than other forms of extracted information, such as word similarity information, local sequential information. To evaluate the different performance by verifying the composition of graph-based methods and discover the best graph-based model, we conduct experiments for graph-based framework with different graph convolution operations and graph pooling methods.

We first evaluate the graph-based methods with different graph convolution operations, feeding the input graph into vanilla GCN and GIN, respectively. From the results of 'GCNs' in the Table 5.1, we can know the GIN outperforms vanilla GCN. The only one difference between GIN and vanilla GCN is that the GIN has an additional MLP layer that can preserve the graph structure comparing with vanilla GCN, therefore, it proved the patient graph structure is important to the prediction once again.

Since GIN can better capture the graph structure information than vanilla GCN, we select GIN as the graph convolution operation in the model architecture, based on which we evaluate the variant graph pooling methods based on SAGPool. In the basic SAGPool method, the authors leverage spectral-based graph convolution operation to calculate the node scoring matrix which can take graph structure into consideration while pooling the graph. From the result of "GIN w/ SAGPool" in the Table 5.1, the Pool(GIN) outperform other pooling methods. The main reason is that the GIN can better preserve the graph structure information and leverage it into pooling step. As a result, the down sampled the graph contains more distinguished nodes preserving the graph structure than other methods.

5.3 Interpretable analysis

The main reason why the graph-based methods are popular recently is that the graph-based methods, such as graph convolutional neural network, can learn the insight information from graph structure where the relationship between nodes are retained in edges. Especially for coarsened graph from well performed graph pooling layers, it is intuitive to interpret the remained patterns of connectivity. Likewise, we can also interpret our best performed framework that is consists of GIN ans GIN-based SAG pooling. The detailed steps of interpretation are as follows:

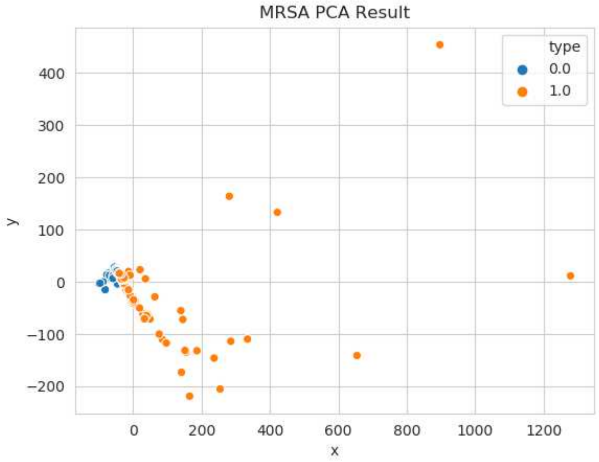


Figure 5.1: PCA result of MRSA truly predicted patients: the orange scatters represent MRSA positive patients and the blue scatters represent MRSA negative patients.

We save the aforementioned best performed model parameters at the point that test auc is 0.8518, and after that, we only use the truly predicted patients from test patient dataset for interpretable analysis among which the number of positive patients is 94 and the number of negative patients is 92. Based on the 186 patients, we first extract the 1 dimensional hidden vectors from the read out layer and feed them into the Principle Component Analysis(PCA). As shown in the Figure 5.1, the MRSA positive patients(orange scatters) and the MRSA negative patients(blue scatters) are generally split into two parts, and the MRSA negative patients are mapped into similar space but MRSA positive patients are not.

Nonetheless, we extract the graphs in the last layer of the SAG pooling for MRSA positive and negative patients, respectively. In order to detect the interpretable patterns, for each class of population, we count the frequency of the node pairs (a.k.a. word co-occurrence pairs) and assign the proportion value

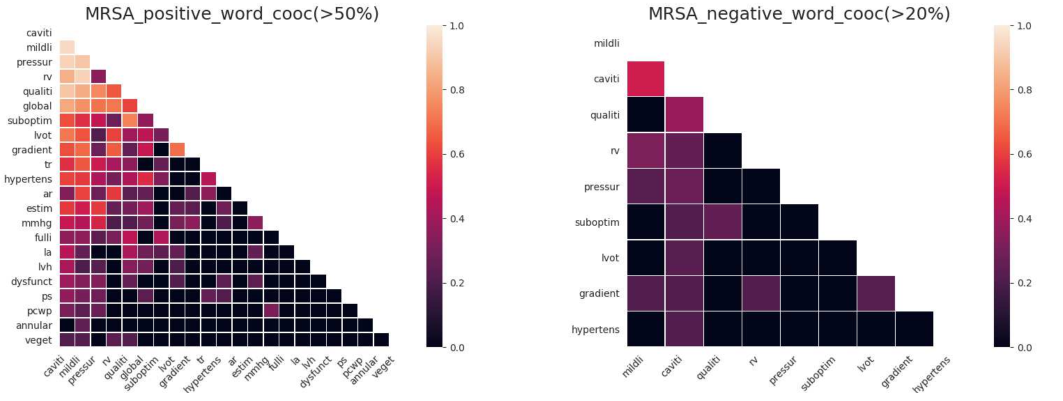


Figure 5.2: Frequent Node Pairs in MRSA Positive(left) and Negative(right) Patients.

for each node pair based on which the node pairs are ranked. As a result, the node pairs whose proportions are more than 20% are filtered and shown in Figure 5.2 for positive and negative patients. In general, the node pairs that meet the conditions in positive patients are more than negative patients and there are dozens of node pairs whose proportions exceed 40% in positive patients but few of such node pair showed in negative patients, both of which indicate that the representative patterns are detected in positive patients and not in negative patients. The interpretation for why the representative patterns can hardly be detected in negative patients is that we randomly subsample patients who have no record of positive infection, which make the negative patient data more diverse and complex than positive patients.

For further interpretable analysis for MRSA positive patients, we extract more representative node pairs whose proportions are more than 50% which can reflect more than half of the populations. As depicted in the Figure 5.3, co-occurrences among more than 10 nodes play the significant role in predicting the MRSA positive patients. Additionally, we investigate the meanings of related words as shown in the Table 5.2. The 'cavity' is a hole that can grow

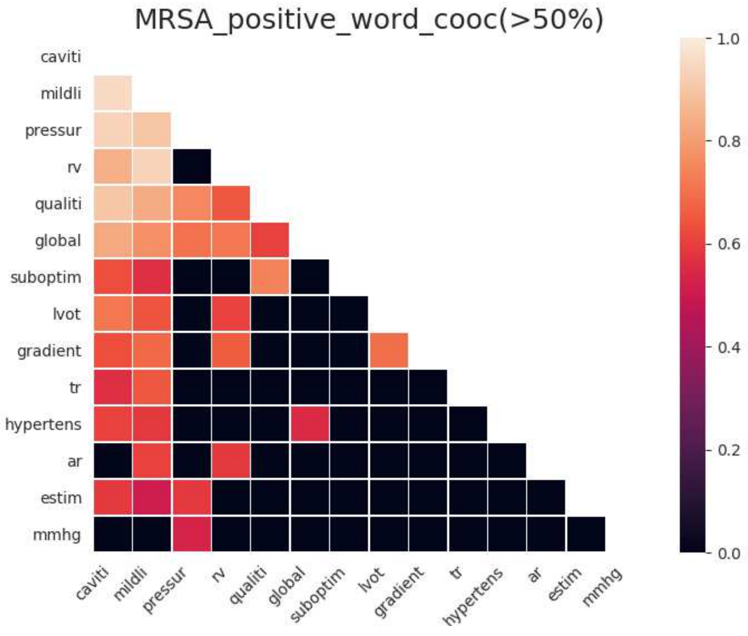


Figure 5.3: Frequent Node Pairs in MRSA Positive Patients(more than 50%).

bigger and deeper over time and MRSA commonly colonized in the nasal cavity. And the 'pressur' seems to denote the pressure ulcer that is the major reservoir of MRSA in hospitals (Pirett *et al.*, 2012). Ganji *et al.* (2019) reported the MRSA pericarditis causing cardiac tamponade from the radiology notes showing that large pericardial effusion with right ventricle. Also Bajraktari *et al.* (2009) reported a case study about left ventricular and mildly thickened mitral valve leaflets. 'tr', 'ar' are also reported in Chesi *et al.* (2006) and Sundaragiri *et al.* (2015). All of the investigations above can indicate that the key words extracted from our framework are meaningful and have relationship with the MRSA.

Since our framework is trained based on graph-level supervised learning, the previous patterns that are extracted from population-based interpretable

Table 5.2: Investigations on key Words extracted from MRSA Positive Patients.

Word	Meaning	reference on MRSA
caviti	Cavity	Colonization of MRSA
mildli	Mildly	Related to radiology notes
pressur	Pressure Ulcers	Pirett *et al.* (2012)
rv	Right Ventricle	Ganji *et al.* (2019)
lvot	Left Ventricular Outflow Tract	Bajraktari *et al.* (2009)
tr	Tricuspid Regurgitation	Chesi *et al.* (2006)
hypertens	Hypertension	Hypertension caused by MRSA
ar	Aortic regurgitation	Sundaragiri *et al.* (2015)

analysis are inherent important, the patient-specific patterns also contribute to the patient-level prediction. We example the graphs distilled in the last SAG pooling layer from 2 MRSA positive patients to show what else nodes and edges are preserved to do the individual-specific predictions. From the Figure 5.4, the patient-specific coarsened graph not only contains the nodes that are extracted from the previous population-based statistics but also contains the nodes with personalized attribute for the patient and both of which contribute to final prediction for the patient.

5.4 Adaboost Result

In order to capture more important information and improve the model performance, we apply the Adaboost algorithm to the framework consisting of GIN and GIN-based SAG pooling that best performed among the all results of variant graph-based frameworks. Each GIN w/ SAGPool(GIN) is regarded

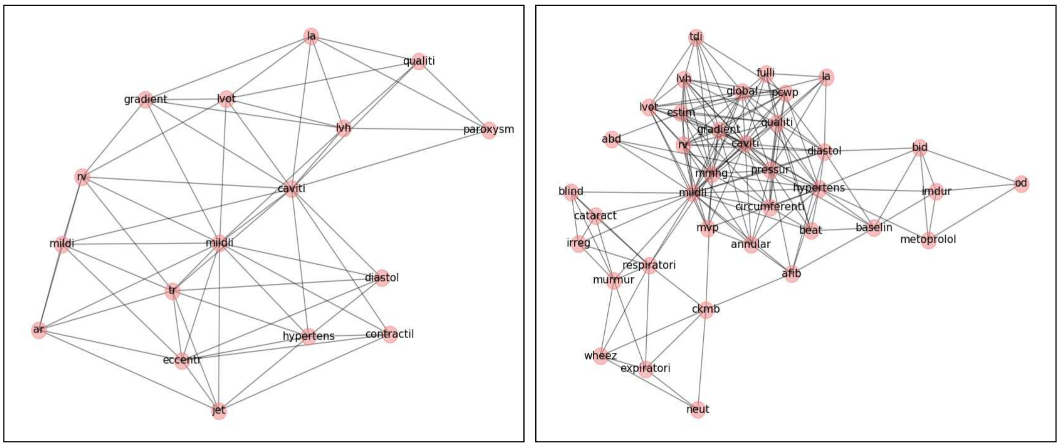


Figure 5.4: Graphs of 2 MRSA positive patients distilled in the last pooling layer.

Table 5.3: Performance comparison of best performed framework without Adaboost algorithm and with Adaboost algorithm(Test-set AUC).

Methods	AUC(mean±std)
GIN w/ SAGPool(GIN)	0.8492±0.0136
Adaboost(GIN w/SAGPool(GIN))	**0.8570±0.0099**

as a weak classification and the number of weak classification is set to 20. Adaboost algorithm weighted sum the results from each weak classification to return a final result, and we also add an annealing learning rate multiplied to the weight with the iteration increasing to converge the model. As depicted in Table 5.3, the Adaboost algorithm combined with our framework has 1∼2% stable improvement, which indicates that the mechanism of the Adaboost algorithm aggregates more distinguished information from different weighted samples by which the more significant patterns could be captured and contribute to better prediction performance.

Chapter 6

Conclusion

In this section, we will sum up the works in this paper and set forth the future works for further improvement.

1. An end-to-end framework for the analysis of clinical notes using graph neural network-base techniques is proposed to predict whether a patient is with MRSA positive infection or negative infection, which can capture the patient-specific and global non-consecutive information from patient clinical notes.

2. Various settings of the GCN-based framework with various combinations of graph convolution operations and graph pooling methods are evaluated, all of which with graph structure information outperformed several baseline methods without graph structure information. Moreover, our framework can easily reach the better performance without the restriction of the note length like sequence-based methods do. Unlike the common deep learning methods, our framework can preserve the graph structure while prediction which makes it interpretable for the results.

3. The framework consisting of GIN as the graph convolution operation and GIN-based self attention graph pooling as the graph pooling method outperformed other frameworks with GCN-based methods, which indicates that learning and preserving the structure information is important to predict the patients.

4. We used the graphs from the pooling layer and leverage them to conduct interpretable analysis from the population-based statistical aspect and patient-specific aspect in MRSA positive patients and MRSA negative patients, respectively.

5. We also apply the Adaboost algorithm that belongs to the ensemble classification to our best performed framework and the framework performance with Adaboost algorithm stably improves 1-2% compared with the framework without Adaboost algorithm.

Since the patient-level graphs are constructed only depending on the frequency information without the time information, it would be studied in the future work. The graph pooling methods still have room to improve which would be another direction in our future work.

Bibliography

Bajraktari, G., Olloni, R., Daullxhiu, I., Ademaj, F., Vela, Z., and Pajaziti, M. (2009). Mrsa endocarditis of bovine contegra valved conduit: a case report. *Cases journal*, **2**(1), 57.

Bastings, J., Titov, I., Aziz, W., Marcheggiani, D., and Sima'an, K. (2017). Graph convolutional encoders for syntax-aware neural machine translation. *arXiv preprint arXiv:1704.04675*.

Berg, R. v. d., Kipf, T. N., and Welling, M. (2017). Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*.

Byrd, R. J., Steinhubl, S. R., Sun, J., Ebadollahi, S., and Stewart, W. F. (2014). Automatic identification of heart failure diagnostic criteria, using text analysis of clinical notes from electronic health records. *International journal of medical informatics*, **83**(12), 983–992.

Chen, S., Varma, R., Sandryhaila, A., and Kovačević, J. (2015). Discrete signal processing on graphs: Sampling theory¡? pub _newline=""? *IEEE transactions on signal processing*, **63**(24), 6510–6523.

Chesi, G., Colli, A., Mestres, C. A., Gambarati, G., Boni, F., and Gherli, T. (2006). Multiresistant-mrsa tricuspid valve infective endocarditis with ancient osteomyelitis locus. *BMC infectious diseases*, **6**(1), 124.

Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. (2016a). Medical concept representation learning from electronic health records and its application on heart failure prediction. *arXiv preprint arXiv:1602.03686*.

Choi, E., Bahadori, M. T., Searles, E., Coffey, C., Thompson, M., Bost, J., Tejedor-Sojo, J., and Sun, J. (2016b). Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1495–1504.

Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. (2017). Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, **24**(2), 361–370.

Choi, E., Xu, Z., Li, Y., Dusenberry, M. W., Flores, G., Xue, Y., and Dai, A. M. (2019). Graph convolutional transformer: Learning the graphical structure of electronic health records. *arXiv preprint arXiv:1906.04716*.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852.

Dhillon, I. S., Guan, Y., and Kulis, B. (2007). Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, **29**(11), 1944–1957.

Douglas, B. L. (2011). The weisfeiler-lehman method and graph isomorphism testing. *arXiv preprint arXiv:1101.5211*.

Dubois, S., Romano, N., Kale, D. C., Shah, N., and Jung, K. (2017). Learning effective representations from clinical notes. *stat*, **1050**, 15.

Ganji, M., Ruiz, J., Kogler, W., Lung, J., Hernandez, J., and Isache, C. (2019). Methicillin-resistant staphylococcus aureus pericarditis causing cardiac tamponade. *IDCases*, **18**, e00613.

Gao, H. and Ji, S. (2019). Graph u-nets. *arXiv preprint arXiv:1905.05178*.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034.

Hartvigsen, T., Sen, C., Brownell, S., Teeple, E., Kong, X., and Rundensteiner, E. A. (2018). Early prediction of mrsa infections using electronic health records. In *HEALTHINF*.

Hastie, T., Rosset, S., Zhu, J., and Zou, H. (2009). Multi-class adaboost. *Statistics and its Interface*, **2**(3), 349–360.

Joffe, E., Pettigrew, E. J., Herskovic, J. R., Bearden, C. F., and Bernstam, E. V. (2015). Expert guided natural language processing using one-class classification. *Journal of the American Medical Informatics Association*, **22**(5), 962–966.

Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific data*, **3**, 160035.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

LeCun, Y., Bengio, Y., *et al.* (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, **3361**(10), 1995.

Lee, J., Lee, I., and Kang, J. (2019). Self-attention graph pooling. *arXiv preprint arXiv:1904.08082*.

Levie, R., Monti, F., Bresson, X., and Bronstein, M. M. (2018). Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, **67**(1), 97–109.

Loper, E. and Bird, S. (2002). Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., and Bronstein, M. M. (2017). Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. (2019). Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4602–4609.

Peng, H., Li, J., He, Y., Liu, Y., Bao, M., Wang, L., Song, Y., and Yang, Q. (2018). Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference*, pages 1063–1072.

Pirett, C., Braga, I. A., Ribas, R., Gontijo Filho, P., and Diogo Filho, A. (2012). Pressure ulcers colonized by mrsa as a reservoir and risk for mrsa bacteremia in patients at a brazilian university hospital. *Wounds*, **24**, 67–75.

Poulin, C., Shiner, B., Thompson, P., Vepstas, L., Young-Xu, Y., Goertzel, B., Watts, B., Flashman, L., and McAllister, T. (2014). Predicting the risk of suicide by analyzing the text of clinical notes. *PloS one*, **9**(1).

Sandryhaila, A. and Moura, J. M. (2013). Discrete signal processing on graphs. *IEEE transactions on signal processing*, **61**(7), 1644–1656.

Sen, C., Hartvigsen, T., Rundensteiner, E., and Claypool, K. (2017). Crest-risk prediction for clostridium difficile infection using multimodal data mining. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–63. Springer.

Sen, C., Hartvigsen, T., Kong, X., and Rundensteiner, E. (2019). Patient-level classification on clinical note sequences guided by attributed hierarchical attention. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 930–939. IEEE.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, **30**(3), 83–98.

Sundaragiri, P. R., Vallabhajosyula, S., Haddad, T. M., and Esterbrooks, D. J. (2015). Tricuspid and mitral endocarditis due to methicillin-resistant staphylococcus aureus exhibiting vancomycin-creep phenomenon. *Case Reports*, **2015**, bcr2015211974.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Vinyals, O., Bengio, S., and Kudlur, M. (2015). Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391*.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*.

Yao, K.-L., Li, W.-J., Yang, J., and Lu, X. (2018). Convolutional geometric matrix completion. *arXiv preprint arXiv:1803.00754*.

Yao, L., Mao, C., and Luo, Y. (2019). Graph convolutional networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., and Leskovec, J. (2018). Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pages 4800–4810.

You, J., Liu, B., Ying, Z., Pande, V., and Leskovec, J. (2018). Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in neural information processing systems*, pages 6410–6421.

Zhang, M., Cui, Z., Neumann, M., and Chen, Y. (2018). An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zitnik, M., Agrawal, M., and Leskovec, J. (2018). Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, **34**(13), i457–i466.

국문초록

전자 건강 기록은 디지털 형태로 체계적으로 수집된 환자의 건강 정보다. 전자 건강 기록이 환자의 상태를 표현 하는 단어들로 구성된 문서의 집합이기때문에 자연어 처리 분야에 적용되는 다양한 기계학습적 방법들이 적용되어왔다. 특히, 딥러닝 기술의 발전으로 인해, 이미지나 텍스트 분야에서 활용 되던 딥러닝 기술들이 생명 정보 및 의학 정보 분야에 점차 적용 되고있다. 하지만, 기존의 이미지나 텍스트데이터와는 다르게, 전자 건강 기록 데이터는 작성자 및 환자 개개인의 상태에 따라서, 데이터의 환자 특이성이 높다. 또한, 유사한 의미를 지니는 건강 기록들간의 상관관계를 고려해야 할 필요가있다. 본연구에서는 전자 건강 기록 데이터의 환자특이성을 고려한 그래프 기반 딥러닝 모델을 고안하였다. 환자의 전자 건강 기록 데이터와 의료 문서들의 공통 출현 빈도를 활용 하여 환자 특이적 그래프를 생성하였다. 이를 기반으로, 그래프 컨볼루션 네트워크를 사용하여 환자의 병리학적상태를예측하는모델을고안하였다. 연구에서 사용한 데이터는 Methicillin-Resistant Staphylococcus Aureus(MRSA) 감염 여부를 측정한 데이터이다. 고안한 그래프기반 딥러닝 모델을 통해 환자의 내성을 예측한 결과, 그래프정보를 활용 하지 않은 기존모델들 보다 2.93%∼11.81% 뛰어난성능을보였다. 또한 해석 가능한 분석을 수행하기 위해 풀링 단계에서 그래프를 조사했다.이를 통해 MRSA 양성 환자에 대해 구별되는 장거리 단어패턴을 찾았으며 환자별 예측에 기여하는 환자의 합동 그래프를 보여 주었다. 성능을 더욱 향상시키기 위해 아다부스트 알고리즘을 사용하였다. 본 논문에서 제안된 결과는 85.70%로 가장 높은 성능을 기록했으며, 이는 기존 모델보다 3.71%∼12.59%의 향상 시켰음을 보여주었다.

Keywords: Clinical notes, Graph Neural Network, Graph Pooling, Interpretable Analysis

Student Number: 2018-27910

감사의 글

I would like to dedicate my paper to all those who have offered me tremendous assistance during the two years in Seoul National University of Computer Science and Engineering.

First of all, my heartiest thanks flow to my supervisor, Professor Kim, for his helpful guidance, valuable suggestions and constant encouragement both in my study and in my life. His profound insight and accurateness about my paper taught me so much that they are engraved on my heart. He provided me with beneficial help and offered me precious comments during the whole process of my writing, without which the paper would not be what it is now.

Also, I would like to express my sincere gratitude to all the members in Bio & Health Informatics Lab, especially Sangseon Lee, who is postdoc fellow in our group and has given me his time in listening to me and helping me work out my problems during the difficult course of the thesis.

Finally, I would like to extend my deep gratefulness to my family whose encouragement and support have made my accomplishments possible.