

Graph-Text Multi-Modal Pre-training for Medical Representation Learning

Sungjin Park¹

Seongsu Bae¹

Jiho Kim¹

Tackeun Kim²

Edward Choi¹

¹KAIST / Daejeon, Republic of Korea

²Department of Neurosurgery, Seoul National University Bundang Hospital / Seongnam, Republic of Korea

ZXZNM@KAIST.AC.KR

SEONGSU@KAIST.AC.KR

JIHO283@KAIST.AC.KR

TACKEUN.KIM@SNU.AC.KR

EDWARDCHOI@KAIST.AC.KR

Abstract

As the volume of Electronic Health Records (EHR) sharply grows, there has been emerging interest in learning the representation of EHR for healthcare applications. Representation learning of EHR requires appropriate modeling of the two dominant modalities in EHR: structured data and unstructured text. In this paper, we present MedGTX, a pre-trained model for multi-modal representation learning of the structured and textual EHR data. MedGTX uses a novel graph encoder to exploit the graphical nature of structured EHR data, and a text encoder to handle unstructured text, and a cross-modal encoder to learn a joint representation space. We pre-train our model through four proxy tasks on MIMIC-III, an open-source EHR data, and evaluate our model on two clinical benchmarks and three novel downstream tasks which tackle real-world problems in EHR data. The results consistently show the effectiveness of pre-training the model for joint representation of both structured and unstructured information from EHR. Given the promising performance of MedGTX, we believe this work opens a new door to jointly understanding the two fundamental modalities of EHR data.

Data and Code Availability This paper utilizes the MIMIC-III dataset (Johnson et al., 2016b), which is available on the PhysioNet repository (Johnson et al., 2016a). Our models and codes are available at the Github repository¹.

1. https://github.com/sjpark9503/kg_txt_multimodal

1. Introduction

The wide adoption of Electronic Health Records (EHR) by healthcare institutions makes it one of the richest sources of medical data, with huge potential for improving digital medicine when combined with machine learning. EHR consists of heterogeneous data modalities and two among them account for a large portion of the data: structured data (*e.g.* medication codes) and unstructured textual data (*e.g.* clinical notes). While the former provides information regarding patient-hospital interaction in a codified and easy-to-process manner, the latter paints a more granular and free-flowing narrative, thus making both inseparable, complementary sources of clinical information (Walsh, 2004).

There have been considerable previous works to learn meaningful representations from EHR by either focusing on each modality or straightforwardly merging them. For structured data understanding, researchers have focused on learning medical concept representations (Choi et al., 2016b, 2017; Zhang et al., 2020) and using the underlying graphical structure of medical concepts (Choi et al., 2018, 2020; Hettige et al., 2020). In terms of textual data understanding, large scale biomedical pre-trained language models (Lee et al., 2020; Alsentzer et al., 2019) have been the dominant approach due to their state-of-the-art performance on biomedical language understanding tasks. Some studies used both structured codes and text for various prediction tasks (Miotto et al., 2016; Suresh et al., 2017; Gong and Gutttag, 2018; Meng et al., 2020), but they treated both codes and text as bag-of-features which ignore the unique nature of each modality. Given that textual and structured data in EHR are not just correlated but also complementary

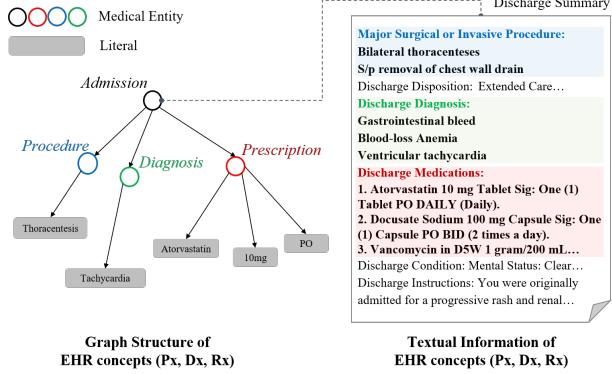


Figure 1: A hospital admission represented in two modalities. **Graph:** the admission node has multiple children nodes, such as procedure (Px), diagnosis (Dx), and prescription (Rx), each with its literal nodes. **Text:** the discharge summary provides another view on the same admission information.

to each other, well-designed joint representation learning will provide the opportunity to better capture the heterogeneous nature of EHR.

In this work, we propose MedGTX, a pre-trained model for joint multi-modal representation learning of the structured and textual EHR data. Specifically, we interpret structured EHR data as a graph (Choi et al., 2018) based on its hierarchical and heterogeneous nature, therefore propose a graph-text multi-modal learning framework. Our approach extends the recently developed two-stream multi-modal learning architectures (Tan and Bansal, 2019; Lu et al., 2019) to the joint representation of graph and text. In addition, we develop a summary network, a novel method to incorporate the textual description of a node into the final graph representation, improving the model performance for all downstream tasks at the cost of small computational overhead. To maximize the model’s multi-modal representation learning for EHR data, we use four different pre-training objectives, namely masked language modeling, masked literal prediction, relation classification, and cross-modal alignment prediction.

Using a publicly available EHR dataset, MIMIC-III (Johnson et al., 2016b), we extensively evaluate the proposed method with various baselines on two clinical benchmarks and three novel downstream tasks that require a multi-modal understanding of both graph and text data in EHR: cross-modal retrieval, error

detection, and clinical note generation. In all downstream tasks, MedGTX outperformed all baselines thanks to the combination of pre-training objectives and the proposed summary network. To our best knowledge, this is the first work to propose graph-text multi-modal pre-training on EHR with rigorous evaluation on various downstream tasks, demonstrating the necessity of a pre-trained model that jointly considers the two important modalities of EHR data. Given the consistent and promising performance of MedGTX, we believe this work can open a new door to jointly understanding the two fundamental modalities of EHR data.

2. Related Works

2.1. Representation Learning for Structured EHR Data

There have been numerous studies focusing on learning representations of various medical concepts such as diagnosis codes, medication orders, and hospital visits.

Early studies leveraged the longitudinal nature of EHR (Tran et al., 2015; Choi et al., 2016b,a; Nguyen et al., 2016) to learn medical code representations from a series of visit records, and further tried to incorporate biomedical domain knowledge (Choi et al., 2017; Ma et al., 2018; Yin et al., 2019; Song et al., 2019; Zhang et al., 2020). More recently, studies have been proposed that try to leverage the underlying structure of EHR, such as focusing on the hierarchical nature between diagnosis and treatment codes (Choi et al., 2018), or the graphical and temporal relationship between multiple medical codes (Choi et al., 2020; Hettige et al., 2020). Note that there have been studies that used both structured codes and text for various prediction tasks (Miotto et al., 2016; Suresh et al., 2017; Gong and Guttag, 2018; Meng et al., 2020). These studies, however, do not fully utilize the unique nature of structured codes or the unstructured text. Structured codes were treated as bag-of-features without considering their complex graphical structure, and the text was aggregated to topics via topic modeling which made it impossible to align words with codes. Although prior works demonstrated successful representation learning for their own tasks, there has not been an attempt to jointly learn the representations of graphical and textual EHR data for various multi-modal downstream tasks.

2.2. Representation Learning for Textual EHR Data

Representation learning for textual healthcare data has been an emerging area of research in recent years. In early studies, pre-trained word embedding methods (Mikolov et al., 2013; Pennington et al., 2014) were widely used to model biomedical texts and applied to various applications, such as named entity recognition (Moen and Ananiadou, 2013), abbreviation disambiguation (Liu et al., 2015), and patient de-identification (Dernoncourt et al., 2017).

Since the success of pre-trained language models (LMs) such as BERT (Devlin et al., 2019), several domain-specific pre-trained LMs have also emerged in the clinical domain. A dominant approach for pre-training is to further train the original BERT over biomedical texts. Specifically for the healthcare field, some LMs (Alsentzer et al., 2019; Peng et al., 2019) utilize clinical texts for pre-training such as clinical notes from MIMIC-III.

2.3. Multi-modal Learning via Pre-Training

Multi-modal representation learning has become one of the promising directions due to richer representation by combining various modalities such as graphs, text, and visual inputs. In vision-language representation learning, a single stream (Su et al., 2019; Chen et al., 2020; Li et al., 2019) and two-stream (Lu et al., 2019; Tan and Bansal, 2019) Transformer-based architecture with pre-training are proposed to jointly represent the image and text. Unlike vision-language representations, previous works on graph-text multi-modal learning attempt to enrich one modality by injecting external information from another modality such as text-aware knowledge embedding (Xu et al., 2017; Xiao et al., 2017; Xie et al., 2016) and knowledge-enhanced language models (Zhang et al., 2019; Peters et al., 2019; Logan et al., 2019). Although there exist some prior works that focus on jointly aligning the embedding of entity and words (Wang et al., 2014; Zhong et al., 2015; Toutanova et al., 2015), they need manual alignment between graph and text and were not evaluated on challenging tasks to verify cross-modal understanding. Recently, similar to our work, GraphCodeBERT (Guo et al., 2021) proposed a model for learning representations of program source codes, where they jointly embed abstract syntax trees and the comments.

In this work, we focus on the graph-text multi-modal learning for EHR by employing four pre-

training tasks and the summary network designed to reflect both the structured and unstructured information in EHR data.

3. Methods

3.1. Model Architecture

As shown in Figure 2, our model employs a two-stream architecture where the model separately encodes each input (graph and text) with self-attention layers and combines two modalities in the cross attention layers. Below, we describe the details of the five main components of MedGTX: the architecture and notations for each modality, attention layer, text encoder, graph encoder, and cross-modal encoder.

3.1.1. TEXT NOTATIONS

Following the text preprocessing and embedding strategy of BERT (Devlin et al., 2019), we first tokenize the input sentence W into N_T tokens (*i.e.* subwords), $W = \{w_1, w_2, \dots, w_{N_T}\}$. We then add special tokens [CLS] and [SEP] at the front and back of W for downstream tasks. Note that our source of text, the discharge summary, consists of multiple sections, and each section is tagged as respective section headers (*e.g.* "Discharge Medication"). As sections can serve a similar role as the segments in BERT, we build a lookup table of section headers $c \in \{0(\text{Dx}), 1(\text{Px}), 2(\text{Rx})\}$. For text embedding, each token w_i , its position i , and section header c_i are transformed to respective embeddings via the embedding layer, then summed up to produce the final embeddings [CLS], $\mathbf{w}_1, \dots, \mathbf{w}_{N_T}$, [SEP].

3.1.2. GRAPH NOTATIONS

Given a EHR graph $G = (V, E)$, $V = \{v_1, \dots, v_{N_v}\}$ is a set of nodes with N_v denoting the number of nodes, and $D = \{d_1, \dots, d_{N_v}\}$ is a set of textual descriptions in a graph where d_i belongs to v_i , and the description consists of N_{d_i} tokens $d_i = w_1^i, \dots, w_{N_{d_i}}^i$ and $E = \{e_1, \dots, e_{N_e}\}$ is a set of relations with N_e denoting the number of relations in the graph G . Our EHR graph is converted from the relational database, therefore some nodes (*i.e.* *abstract nodes*) represent abstract concepts such as a single admission or a single medication order, while other nodes (*i.e.* *literal nodes*) represent concrete values such as a medication volume ("250mg"). Based on this observation, we use special tokens ([ADM], [DX], [PX], [RX]) to represent the

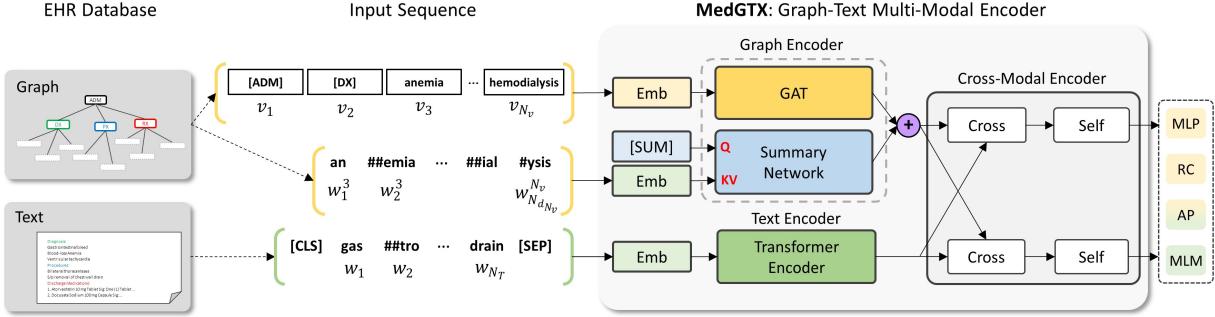


Figure 2: Overview of the proposed MedGTX model for the joint representation learning of structured and unstructured data in EHR. The model consists of two uni-modal encoders and the cross-modal encoder. We pre-train the model with four objectives: masked language modeling (MLM), masked literal prediction (MLP), relation classification (RC), and cross-modal alignment prediction (AP).

abstract nodes while literal nodes are represented with unique embeddings. Furthermore, abstract nodes’ d_i (*e.g.* admission ID) are unused throughout the paper as they provide little information to the model. The difference between abstract nodes and literal nodes are further discussed in Appendix A.1.

For graph embedding, each node v_i and its position i are transformed to respective embeddings via the embedding layer, then summed up to produce the final embeddings $\mathbf{v}_1, \dots, \mathbf{v}_{N_v}$. Note that position embeddings are required for the model to distinguish multiple abstract nodes of the same type (*e.g.* a single admission contains multiple medication orders).

3.1.3. MULTI-HEAD ATTENTION LAYER

We define the multi-head attention mechanism (Vaswani et al., 2017) which will be used in all uni-modal and cross-modal encoders. We use the canonical scaled dot-product as follows, where we omit the multi-head component to reduce clutter,

$$\begin{aligned} \mathbf{Q} &= \mathbf{H}_Q \mathbf{W}_Q, \quad \mathbf{K} = \mathbf{H}_{KV} \mathbf{W}_K, \quad \mathbf{V} = \mathbf{H}_{KV} \mathbf{W}_V \\ \text{Attn}(\mathbf{H}_Q, \mathbf{H}_{KV}, \mathbf{M}) &= \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V} \end{aligned}$$

where \mathbf{H}_Q denotes the input features for queries, \mathbf{H}_{KV} the input features for keys and values, d_k the dimension size of the key vectors, d_v the dimension size of the value vectors, and d_h the dimension size of hidden states. $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d_h \times d_k}$, and $\mathbf{W}_V \in \mathbb{R}^{d_h \times d_v}$ are trainable projection weights for query, key, and value. \mathbf{M} denotes an attention mask comprised of 0s and

negative number ns for indicating which features are allowed to attend to one another.

3.1.4. TEXT ENCODER

The text encoder consists of L_T layers of Transformer encoders, where the first layer takes text embeddings $\mathbf{H}_T^{(0)} = [\text{CLS}], \mathbf{w}_1, \dots, \mathbf{w}_N, [\text{SEP}]$. Transformer encoders are then repeatedly applied, which consist of the multi-head self-attention and the position-wise feed-forward network. The output $\mathbf{H}_T^{(l)}$ of the l -th text encoder is obtained as follows,

$$\mathbf{H}_T^{(l)} = \text{Attn}(\mathbf{H}_T^{(l-1)}, \mathbf{H}_T^{(l-1)}, \mathbf{M}_{pad})$$

where \mathbf{M}_{pad} consists of ns at padding token positions to exclude it from attention score, and 0s elsewhere.

3.1.5. GRAPH ENCODER

The graph encoder consists of L_G layers of two separate components: the graph attention network (GAT) (Veličković et al., 2018) and the summary network. GAT encodes the graphical structure of the EHR graph by considering the adjacency between all pairs of nodes. Starting with $\mathbf{H}_G^{(0)} = \mathbf{v}_1, \dots, \mathbf{v}_{N_v}$, at l -th GAT layer, the previous output $\mathbf{H}_G^{(l-1)}$ is transformed as follows,

$$\mathbf{H}_G^{(l)} = \text{Attn}(\mathbf{H}_G^{(l-1)}, \mathbf{H}_G^{(l-1)}, \mathbf{M}_{adj})$$

where \mathbf{M}_{adj} is a symmetric adjacency matrix that reflects the EHR graph structure.

A common approach to introduce node features D into GAT is initializing each node embedding \mathbf{v}_i with the mean of token embeddings of d_i . However, EHR graphs often contain repetitive and redundant words (*e.g.* “main”, “IV”) in d_i , and the importance of each word w_j^i depends on the downstream task. To this end, we propose a novel summary network that extracts keywords from D and encodes them into a fixed-size summary vector \mathbf{s} independent of GAT. We add a special token [SUM] to the node lookup table and its embedding vector is used as the input $\mathbf{s}^{(0)}$ to the summary network. At l -th layer of the summary network, the previous output $\mathbf{s}^{(l-1)}$ is transformed as follows,

$$\mathbf{s}^{(l)} = \text{Attn}\left(\mathbf{s}^{(l-1)}, \mathbf{H}_D, \mathbf{M}_{\text{pad}}\right)$$

where $\mathbf{H}_D = \mathbf{w}_1^1, \mathbf{w}_2^1, \dots, \mathbf{w}_{N_{d_{N_v}}}^{N_v}$ is a concatenation of all descriptions in the graph. Note that the purpose of the summary network is the extraction of keywords rather than contextualization of the unstructured node features in the graph, so we directly inject the low-level text features as a key and value of the summary network. We empirically demonstrate the superiority of our summary network over the node initialization technique in Section 5.1.

3.1.6. CROSS-MODAL ENCODER

We apply the cross-modal encoder on the text features $\mathbf{H}_T^{(L_T)}$ and the graph features $\mathbf{H}_G^{(L_G)}$ and $\mathbf{s}^{(L_G)}$. The cross-modal encoder (Tan and Bansal, 2019) consists of L_C cross-attention layers, and L_S self-attention layers, and position-wise feed-forward networks. The cross-attention is equivalent to the encoder-decoder attention (Vaswani et al., 2017), in that the key and value vectors come from the other modality (Fig. 2). The output of the l -th cross-attention layer is,

$$\begin{aligned} \mathbf{O}_T^{(l)} &= \text{Attn}\left(\mathbf{O}_T^{(l-1)}, \mathbf{O}_G^{(l-1)}, \mathbf{M}_{\text{pad}}\right) \\ \mathbf{O}_G^{(l)} &= \text{Attn}\left(\mathbf{O}_G^{(l-1)}, \mathbf{O}_T^{(l-1)}, \mathbf{M}_{\text{pad}}\right) \end{aligned}$$

where $\mathbf{O}_T^{(0)} = \mathbf{H}_T^{(L_T)}$, $\mathbf{O}_G^{(0)} = [\mathbf{s}^{(L_G)}; \mathbf{H}_G^{(L_G)}]$, and $\mathbf{O}_T^{(l)}$ is a text output which is blended with graph features, $\mathbf{O}_G^{(l)}$ vice versa. Note that $[\mathbf{s}^{(L_G)}; \mathbf{H}_G^{(L_G)}]$ is a concatenation of $\mathbf{s}^{(L_G)}$ and $\mathbf{H}_G^{(L_G)}$ along the sequence. Here we assume that the summary network was used. We use $\mathbf{s}^{(0)}$ instead of $\mathbf{s}^{(L_G)}$ when we omit the summary network, which is similar to [CLS] in the text modality. The process for the self-attention layers and

Table 1: Data statistics of Dx+Px and Rx. Train/Valid/Test split for # samples. # nodes is the sum of abstract nodes and literal nodes.

Dataset	# samples	# nodes	# relations	# literals
Dx+Px	28,915 / 2,000 / 2,000	699,724	6	8,049
Rx	23,663 / 1,000 / 1,000	3,015,345	7	9,291

the feed-forward networks are the same as uni-modal encoders.

3.2. Pre-training Objectives

We describe four pre-training objectives used for training MedGTX. Further pre-training details can be found in Appendix C.1.

Masked Language Modeling (MLM) We use the masked language modeling (MLM) task from BERT to pre-train for the text modality. A random sample of the tokens in the text input sequence is selected and replaced with the special token [MASK]. The MLM objective is a cross-entropy loss on predicting the masked tokens. Unlike BERT, MedGTX is encouraged to leverage both textual context and graph information to recover the masked tokens correctly. During pre-training, words in the input sequence are randomly masked with a probability of 0.15. We replace the chosen token with (1) the [MASK] token with 80% chance (2) a random token with 10% chance (3) the original token with 10% chance.

Masked Literal Prediction (MLP) Similar to MLM, we randomly mask literal nodes of the EHR graph with a probability of 0.15 and pre-train the model to recover its identity as another way to encourage MedGTX to leverage both structured and unstructured modalities. Unlike MLM, we do not replace the nodes with a random node since the graph encoder considers the node connections.

Relation Classification (RC) As there are multiple relation types in the EHR graph (*e.g.* *MedicationOrder*, *Diagnosis*, *DiagnosisName*, *DrugRoute*), which were not explicitly injected into the model architecture, we propose relation classification to learn the multi-relational aspect of structured EHR. Specifically, we randomly sample 10% of node pairs in the graph, concatenate their corresponding output contextualized features, and predict their relation type, including *Not Connected*.

Cross-modal Alignment Prediction (AP) We follow the cross-modal alignment prediction task in LXMERT (Tan and Bansal, 2019). The purpose of the cross-modal alignment prediction objective is to encourage the model to align a representation space of the graph and language modality. First, for each graph-text pair, we replace it with a text from another admission with the probability of 0.5. Then, we concatenate the special tokens [SUM] and [CLS] to classify whether the given graph-text pair comes from the same admission or not.

4. Experiments

In this section, we first introduce the dataset used for pre-training and downstream evaluation, followed by the baseline model description. We then describe five downstream tasks: two existing clinical benchmarks and three novel multi-modal downstream tasks, along with experimental results. A detailed description of experimental settings for each downstream task can be found in Appendix C.

4.1. Datasets

We evaluate all models on MIMIC-III (Johnson et al., 2016b), a freely-available multi-modal EHR dataset. Collected for approximately 40,000 ICU patients at Beth Israel Deaconess Medical Center between 2001 and 2012, MIMIC-III consists of heterogeneous data such as diverse structured information (*e.g.* diagnoses, prescriptions, procedures) and textual data (*e.g.* physician notes, nursing notes, discharge summaries). We convert MIMIC-III to a graph from a relational database by following the approach in MIMICSQL* (Park et al., 2021). We describe the details of our EHR graph in Appendix B. Specifically for text data, we use discharge summaries, which contain a summary of the medical events that occurred during single hospital admission (Figure 1). Given that discharge summaries are divided into sections , we can extract text regarding the diagnoses (Dx), procedures (Px), and prescriptions (Rx) that occurred during the admission. As an initial step towards applying our framework to the entire EHR, we merge datasets which share a certain level of similarity. Specifically, Dx and Px both share the exactly same table schema in MIMIC-III. Furthermore, the number of records in Dx and Px are comparatively smaller than Rx. Therefore, we merge Dx and Px into a single dataset Dx+Px without changing the size and characteris-

tics. We also considered merging Dx, Px, and Rx into a single dataset, Dx+Px+Rx, but this would significantly increase the text length and the number of nodes in a graph. Due to the limitation of computational resource, we keep the prescriptions as a separate dataset Rx, and leave the development of an efficient approach to encode a large graph and text from EHR as a future work. A summary statistics of the two datasets is provided in Table 1.

4.2. Models

- **Uni-modal baselines** To verify the effectiveness of the multi-modal joint representation, we evaluate uni-modal baselines on downstream tasks. For the text modality, we employ a biomedical pre-trained language model, ClinicalBERT (Alsentzer et al., 2019), which is trained on PubMed and MIMIC-III. For the graph modality, we employ GAT (Veličković et al., 2018) (or Transformer encoder) to encode the graph with (or without) the structural information, respectively. Furthermore, we also compare with SAnD (Song et al., 2018) which interprets our graph as a time series. We evaluate SAnD only on Rx because Dx+Px does not have temporal information. For the note generation task, we choose a simple sequence-to-sequence (Seq2Seq) as the uni-modal baseline, which encodes the graph with a Transformer encoder and decodes the text with a Transformer decoder.
- **Concat** We replace cross attention layers in MedGTX with self-attention layers, equivalent to two independent single modality encoders. For a fair comparison, Concat is pre-trained with the same objectives as MedGTX. This baseline is used to verify the effect of the joint representation learning of graph and text modalities.
- **MedGTX** Both MedGTX and MedGTX+summary are our pre-trained models, but the graph encoder of MedGTX does not contain the summary network proposed in Section 3.1.4. - PT is proposed to verify the effectiveness of the pre-training, where we randomly initialize MedGTX and directly fine-tune it for downstream tasks.

In all downstream tasks, all models were trained with five random seeds, and we report the mean and the standard deviation. We use all four pre-training tasks for the Dx+Px dataset, while we exclude the relation classification (RC) task for the Rx dataset as all

Table 2: Test-set results on text & graph retrieval. We report Hits@10 and mean reciprocal rank.

Model	Text Retrieval				Graph Retrieval			
	Dx+Px		Rx		Dx+Px		Rx	
	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR	Hits@10	MRR
Concat	51.6 (0.3)	24.1 (0.4)	74.0 (2.9)	43.3 (4.2)	52.0 (0.8)	24.8 (0.5)	74.8 (0.7)	45.5 (1.5)
MedGTX	76.1 (0.6)	47.2 (0.3)	93.0 (0.2)	75.8 (1.4)	76.4 (0.5)	47.8 (0.4)	92.1 (0.6)	77.4 (1.1)
- PT	2.0 (0.4)	1.3 (0.2)	1.0 (0.0)	0.8 (0.0)	2.0 (0.7)	1.3 (0.3)	1.1 (0.1)	0.8 (0.1)
+ summary	77.4 (0.3)	51.4 (0.6)	96.2 (0.1)	87.7 (0.1)	78.2 (0.4)	52.8 (0.4)	95.5 (0.1)	86.9 (0.1)

Table 3: Test-set results on mortality prediction. We report Area Under Precision-Recall Curve.

Modality	Model	Dx+Px	Rx
Uni-Modal	ClinicalBERT	12.3 (1.8)	37.8 (3.4)
	GAT	20.0 (2.3)	26.7 (4.1)
	Transformer	19.6 (1.4)	25.8 (3.4)
	SAnD	-	10.0 (2.1)
Multi-Modal	Concat	19.5 (1.1)	41.4 (1.4)
	MedGTX	21.2 (0.5)	43.2 (2.5)
	- PT	19.2 (1.4)	37.1 (0.7)
	+ summary	24.1 (0.8)	43.9 (1.5)

models showed better performance without it. Also, we initialize the text encoder with parameters from BERT-tiny (Turc et al., 2019).

4.3. Cross-Modal Retrieval

Cross-modal retrieval aims to find the most relevant one among a pool of text given a graph (*i.e.* Text Retrieval) or vice versa (*i.e.* Graph Retrieval). This task serves as a sanity check for the cross-modal alignment objective. At inference, given pairs are ranked based on their respective prediction scores. We use Hits@10 and mean reciprocal rank (MRR) to report the model performance for both text retrieval and graph retrieval. Hits@k describes the fraction of true graph (or text) that appear in the first k graphs (or text) of the sorted rank list, and MRR is the inverse of the harmonic mean of the ranks.

Table 2 shows the model performance on the Dx+Px and Rx datasets. Results show that for all models, the proposed pre-training objectives dramatically increase the retrieval performance for all cases. The summary network also leads to a dramatic increase of MRR in both retrieval tasks compared to MedGTX, indicating the effectiveness of the proposed model architecture.

Table 4: Test-set results on readmission prediction. We report Area Under Precision-Recall Curve.

Modality	Model	Dx+Px	Rx
Uni-Modal	ClinicalBERT	32.2 (1.6)	34.8 (3.0)
	GAT	36.5 (1.8)	37.4 (2.3)
	Transformer	35.8 (1.3)	35.0 (3.2)
	SAnD	-	28.6 (1.1)
Multi-Modal	Concat	36.1 (1.1)	41.0 (2.3)
	MedGTX	38.4 (0.7)	38.0 (1.5)
	- PT	37.0 (1.3)	34.2 (2.3)
	+ summary	39.6 (0.8)	41.9 (1.4)

4.4. Temporal Prediction

Predicting mortality and hospital readmission are two of the most common clinical application tasks, as they may improve efficiency and reduce the burden of clinical experts. The prediction target of mortality prediction is either dead (1) or alive (0), and either readmitted (1) or not (0) for readmission prediction. The prevalence rate of mortality and readmission is 4% and 27%, respectively. We construct two benchmark datasets based on MIMIC-III and report the area under the precision-recall curve (AUPRC) as a performance metric. Note that MedGTX is trained with discharge summaries, so if we predict in-hospital mortality, that would be leaking information from the future. Therefore, instead, we set up the mortality prediction task to predict whether the patient would die within 30 days after discharge (similar to predicting the 30-day readmission).

Table 3 and 4 show the performance of mortality and readmission prediction, respectively. We can observe the trend that using both structured and unstructured modalities generally improves AUPRC compared to uni-modal baselines, and that our pre-training objectives indeed are helpful for clinical pre-

Table 5: Test-set results on error detection. We report F1 score.

Modality	Model	Dx+Px	Rx
Uni-Modal	GAT	80.0 (0.2)	89.6 (0.1)
	Concat	84.1 (0.3)	99.1 (0.0)
	MedGTX	84.3 (0.3)	99.1 (0.1)
Multi-Modal	- PT	83.8 (0.3)	95.9 (0.1)
	+ summary	84.6 (0.2)	99.1 (0.1)

dictions. Furthermore, the description summary of a graph in MedGTX has an advantage even when the task requires prediction over time dimension.

4.5. Error Detection

EHR often contains misspelled values which can lead to critical situations, therefore detecting such errors in advance is an important application. As structured values are often written based on clinical notes, a chance of error is higher for the graph modality than the text. Therefore, given a graph-text pair, we replace some literal nodes with another random literal node and train all models to predict if any node was altered. Since finding as many errors as possible with high precision is the objective of this task, we use F1 score to report the model performance. Table 5 shows the error detection performance on the Dx+Px and Rx datasets. Again, it can be seen that the multi-modal signal improves performance for all models in both datasets, showing error detection also makes use of clues from another modality. However, given the similar performance of Concat and MedGTX, we can infer that a simple multi-modal pre-training task is enough to produce good performance even without explicit cross-attention.

We discover that most model performances are saturated to a similar point. We believe this is due to the random replacement of nodes in a graph yielding some trivial errors, and leave the generation of non-trivial errors as the future work.

4.6. Clinical Note Generation

Clinical note generation aims to translate EHR graph into clinical free text, similar to image captioning in the vision-language domain. Compared to other downstream tasks, this one evaluates a different aspect of multi-modal learning, namely the generation capability. Since generation tasks require a decoder, we adopt

Table 6: Test-set results on clinical note generation. We report ROUGE-2/L F1 score.

Modality	Model	Dx+Px		Rx	
		RG-2	RG-L	RG-2	RG-L
Uni-Modal	Seq2Seq	6.8 (0.1)	20.2 (0.1)	33.0 (0.4)	52.8 (0.3)
	MedGTX	7.4 (0.1)	20.6 (0.1)	43.8 (0.2)	54.1 (0.2)
Multi-Modal	- PT	3.3 (0.2)	13.9 (0.5)	34.4 (0.4)	44.5 (0.7)
	+ summary	7.5 (0.1)	20.8 (0.2)	44.1 (0.2)	54.7 (0.2)

a UniLM style (Dong et al., 2019) fine-tuning and decoding strategy to train all models to generate text without changing the overall model architecture. We adopt ROUGE (RG) score as the evaluation metric where RG-2 measures the overlap of bigrams and RG-L measures the longest common subsequence (LCS) between the system output and reference. As shown in Table 6, both datasets enjoy significantly improved performance thanks to our pre-training objectives, and further smaller performance boost thanks to the summary network.

5. Analysis

In this section, we analyze MedGTX from the perspective of some alternative design choices. We then perform ablation studies over model components and pre-training objectives. We also qualitatively assess cross-modal attention and analyze its clinical meaning. We visualize additional qualitative samples in Appendix D.

5.1. Ablation Studies

5.1.1. EFFECT OF ALTERNATIVE DESIGN CHOICES

We describe four variants of MedGTX and report their downstream task performances in Table 7.

+ **init** is another way of infusing textual descriptions into the graph representation. + **init** replaces the graph embeddings with the mean of the textual description token embeddings. In general, the summary network shows better performance, but + **init** outperforms on cross-modal retrieval. Unlike MedGTX+summary, + **init** has the advantage to capture the bag-of-(sub)words similarity between two modalities because + **init** directly injects the bag-of-words as input and keeps it along with a residual connection. Despite the importance of the bag-of-words similarity in the information retrieval, our model offers a flexible summary of textual descriptions in a

Table 7: Ablation over design choices. We report AUPRC for temporal predictions, F1 for error detection, ROUGE-L for generation, and mean reciprocal rank (MRR) for text retrieval.

Model	Mortality		Readmission		Error Detection		Generation		Text Retrieval	
	Dx+Px	Rx	Dx+Px	Rx	Dx+Px	Rx	Dx+Px	Rx	Dx+Px	Rx
MedGTX + summary	21.2 (0.5)	43.2 (2.5)	38.4 (0.7)	38.0 (1.5)	84.3 (0.3)	99.1 (0.1)	20.6 (0.1)	54.1 (0.2)	47.2 (0.3)	75.8 (1.4)
	24.1 (0.8)	43.9 (1.5)	39.6 (0.8)	41.9 (1.4)	84.6 (0.2)	99.1 (0.1)	20.8 (0.2)	54.7 (0.2)	51.4 (0.6)	87.7 (0.1)
	22.4 (0.8)	39.9 (1.4)	37.2 (1.6)	38.5 (2.3)	81.9 (0.4)	96.6 (0.1)	21.2 (0.1)	54.3 (0.2)	59.7 (0.4)	89.2 (0.3)
	- structure	21.8 (0.6)	41.5 (1.1)	36.2 (1.9)	40.0 (1.8)	-	-	20.1(0.2)	56.3 (0.4)	41.2 (2.0)

Table 8: Ablation over model components. We report AUPRC for temporal predictions, F1 for error detection, ROUGE-L for generation, and mean reciprocal rank (MRR) for text retrieval.

Model	Mortality		Readmission		Error Detection		Generation		Text Retrieval	
	Dx+Px	Rx								
MedGTX+summary - GAT	24.1 (0.8)	43.9 (1.5)	39.6 (0.8)	41.9 (1.4)	84.6 (0.2)	99.1 (0.1)	20.8 (0.2)	54.7 (0.2)	51.4 (0.6)	87.7 (0.1)
	22.5 (2.3)	44.1 (1.5)	36.4 (1.5)	37.8 (1.5)	84.3 (0.3)	96.6 (0.1)	20.4 (0.2)	54.9 (0.2)	36.6 (0.1)	80.9 (0.3)
	22.4 (0.9)	40.0 (1.7)	36.2 (1.2)	40.9 (1.2)	84.5 (0.2)	96.6 (0.1)	20.1 (0.1)	50.7 (0.2)	35.5 (1.0)	62.0 (0.5)

graph, which leads to a performance improvement on other tasks.

- structure is used to test the importance of the structural information of EHR graph. We destroy the structural information by concatenating all textual description D into a single passage to represent the EHR graph. As there is no structure we use Transformer instead of GAT to encode the passage. For both Dx+Px and Rx, the structural information of a graph plays a crucial role for all tasks except note generation for Rx. This can be due to the Rx section of the discharge summary being more or less a permutation of textual descriptions D , making the note generation a rather straightforward problem for **- structure**. However, **- structure** loses merit when the task is hard to solve with only textual descriptions D , such as generation on Dx+Px.

MedGTX+summary shows superior performance on three tasks and takes second place on the retrieval and generation. Given MedGTX+summary’s stable performance in all tasks, we believe MedGTX+summary has a larger potential for generalizability than other baselines.

5.1.2. EFFECT OF MODEL COMPONENTS

Our design choices were motivated by a volume of recent multi-modal frameworks from other domains such as vision-language or code-text representation learning (Tan and Bansal, 2019; Guo et al., 2021). To empirically justify our multi-modal framework, we evaluate the effect of two model components, namely the structure-aware attention mask and the text encoder initialization with BERT parameters. Here, we

employ MedGTX+summary as a core model, which shows the impressive performance among the variants of our multi-modal framework. For **- GAT**, we replace the structure-aware attention mask M_{adj} with the padding mask M_{pad} of the graph encoder. For **- LM**, we randomly initialize the text encoder instead of using the pre-trained weights from BERT. We report the downstream task performance in Table 8. Note that the performance of the error detection on Rx is saturated for all models, so we exclude the results from Table 8.

In all downstream tasks, the performance significantly drops when we do not initialize MedGTX+summary with BERT, so the pre-trained language model plays a crucial role in understanding the discharge summary. In the case of **- GAT**, it shows a comparable performance for mortality prediction on Rx, error detection on Dx+Px, and note generation while other task performances significantly decrease. MedGTX+summary always experiences the performance drop when we remove one of the components from the model, indicating the effectiveness of the proposed model architecture.

5.1.3. EFFECT OF PRE-TRAINING TASKS

We establish masked language modeling (MLM) for the text part and masked literal prediction (MLP) for the graph part as major pre-training tasks. Therefore, we study the effect of the other two pre-training tasks, namely cross-modal alignment prediction (AP) and relation classification (RC), on the downstream tasks as shown in Table 9. For this experiment, we use MedGTX+summary as a base model. Two ad-

Table 9: Ablation over pre-training tasks. We report AUPRC for temporal predictions, F1 for error detection, ROUGE-L for generation, and mean reciprocal rank (MRR) for text retrieval.

Pre-training task	Mortality		Readmission		Error Detection		Generation		Text Retrieval	
	Dx+Px	Rx	Dx+Px	Rx	Dx+Px	Rx	Dx+Px	Rx	Dx+Px	Rx
w/ AP, RC	24.1 (0.8)	44.6 (1.3)	39.6 (0.8)	37.5 (1.9)	84.6 (0.2)	20.8 (0.2)	53.4 (0.1)	51.4 (0.6)	56.0 (1.1)	
w/ AP	22.9 (1.7)	43.9 (1.5)	37.1 (1.2)	41.9 (1.4)	84.4 (0.2)	20.9 (0.2)	54.7 (0.2)	46.0 (0.3)	87.7 (0.1)	
w/ RC	22.2 (1.0)	39.4 (1.8)	40.8 (2.0)	40.8 (4.0)	84.7 (0.2)	20.6 (0.3)	53.8 (0.3)	14.0 (1.3)	13.6 (2.9)	
-	24.8 (0.6)	39.9 (1.4)	38.6 (1.4)	40.1 (1.5)	83.9 (0.4)	20.7 (0.1)	54.0 (0.3)	13.0 (0.4)	28.0 (2.9)	

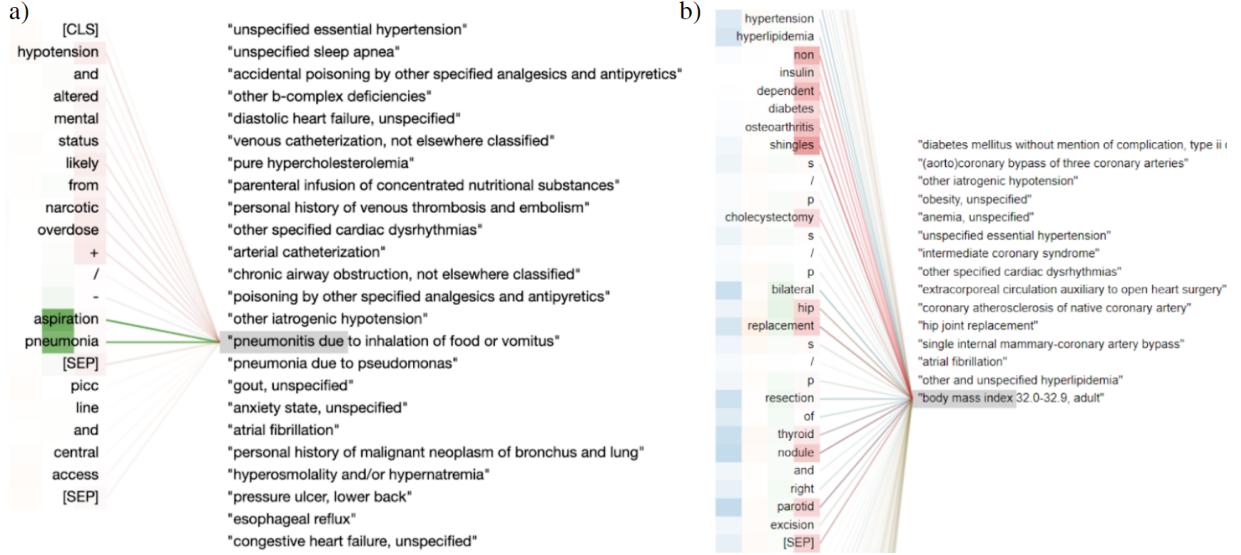


Figure 3: Examples of the cross-modal attention distribution where the query is a graph node and the key is a text token. The left and right sides of each figure are text tokens and graph nodes, respectively. Different colors represent different attention heads. Best viewed in color.

ditional pre-training objectives consistently show a positive impact on most datasets and downstream tasks except for mortality prediction on Rx, but their effect varies across datasets and tasks. In the case of the Rx dataset, MedGTX+summary trained with AP consistently shows a comparable and outperforming performance on every downstream task. For the Dx+Px dataset, training MedGTX+summary with AP and RC is a reasonable choice due to its stable performance over downstream tasks. Furthermore, it can be seen that AP is essential for the cross-modal alignment task.

5.2. Attention Visualization

To indirectly evaluate the model’s understanding of the multi-modal input, we can observe how one modality refers to another in the cross-attention layer. We

observe the three types of connection between two modalities: 1) same surface form, 2) different surface forms, but represent the same medical concept, and 3) different medical concepts, but capture a biomedical correlation between concepts. We visualize the attention weights in Figure 3. In Figure 3 a), the literal node *pneumonitis due to inhalation of food or vomitus* strongly focuses on *aspiration pneumonia* in text. This implies that cross-attention helps to align the same clinical concepts even they have a different surface form. In Figure 3 b), literal node *body mass index 32.0-32.9, adult* not only refers to its synonym but also other clinically meaningful texts such as *hyperlipidemia*, *diabetes*, and *osteoarthritis*, which are closely related to the lifestyle disease (*i.e.*, adult diseases). Other examples and detailed explanations are in Appendix D.

6. Conclusion

This paper presents MedGTX, which learns joint representations of the structured and textual data of EHR. To the best of our knowledge, this is the first pre-trained model that treats both modalities equally compared to previous approaches that handle one modality as an auxiliary information source of another modality. Focusing on the EHR specific nature, we proposed four pre-training tasks and a novel summary network which brought improved performance for the five clinical downstream tasks. Further qualitative analysis and ablation study confirmed the efficacy of the proposed methods. One important future direction is to extend our model to entire EHR data (not just discharge summaries), which will lead to a true understanding of EHR, although extensive pre-processing and hardware resource requirements must be addressed at the same time.

Institutional Review Board (IRB)

This research does not require IRB approval.

Acknowledgments

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST)), National Research Foundation of Korea (NRF) grant (NRF-2020H1D3A2A03100945), and the Korea Health Industry Development Institute (KHIDI) grant (No.HI21C1138), funded by the Korea government (MSIT, MOHW).

References

- Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, 2019.
- Nilesh Chakraborty, Denis Lukovnikov, Gaurav Mapheshwari, Priyansh Trivedi, Jens Lehmann, and Asja Fischer. Introduction to neural network-based question answering over knowledge graphs. *WIREs Data Mining and Knowledge Discovery*, 11(3):e1389, 2021.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.
- Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. 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, 2016a.
- Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare representation learning. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 787–795, 2017.
- Edward Choi, Cao Xiao, Walter F Stewart, and Jimeng Sun. Mime: multilevel medical embedding of electronic health records for predictive healthcare. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 4552–4562, 2018.
- Edward Choi, Zhen Xu, Yujia Li, Michael Dusenberry, Gerardo Flores, Emily Xue, and Andrew Dai. Learning the graphical structure of electronic health records with graph convolutional transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 606–613, 2020.
- Youngduck Choi, Chill Yi-I Chiu, and David Sontag. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41, 2016b.
- Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606, 2017.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32, pages 13063–13075. Curran Associates, Inc., 2019.
- Jen J Gong and John V Guttag. Learning to summarize electronic health records using cross-modality correspondences. In *Machine Learning for Healthcare Conference*, pages 551–570. PMLR, 2018.
- Daya Guo, Shuo Ren, Shuai Lu, Zhangyin Feng, Duyu Tang, Shujie Liu, Long Zhou, Nan Duan, Alexey Svyatkovskiy, Shengyu Fu, et al. Graphcodebert: Pre-training code representations with data flow. In *ICLR*, 2021.
- Bhagya Hettige, Weiqing Wang, Yuan-Fang Li, Suong Le, and Wray Buntine. Medgraph: structural and temporal representation learning of electronic medical records. In *European Conference on Artificial Intelligence 2020*, pages 1810–1817. IOS Press, 2020.
- Alistair E. W. Johnson, Tom J. Pollard, and Roger G. Mark. MIMIC-III clinical database (version 1.4), 2016a.
- Alistair EW Johnson, Tom J Pollard, Lu Shen, Liwei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016b.
- Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.
- Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.
- Yue Liu, Tao Ge, Kusum S Mathews, Heng Ji, and Deborah L McGuinness. Exploiting task-oriented resources to learn word embeddings for clinical abbreviation expansion. *ACL-IJCNLP 2015*, page 92, 2015.
- Robert Logan, Nelson F Liu, Matthew E Peters, Matt Gardner, and Sameer Singh. Barack’s wife hillary: Using knowledge graphs for fact-aware language modeling. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
- Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- Fenglong Ma, Quanzeng You, Houping Xiao, Radha Chitta, Jing Zhou, and Jing Gao. Kame: Knowledge-based attention model for diagnosis prediction in healthcare. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 743–752, 2018.
- Yiwen Meng, William Farran Speier, Michael Ong, and Corey Arnold. Hct: Hierarchical clinical embedding with topic modeling on electronic health record for predicting depression. *IEEE Journal of Biomedical and Health Informatics*, 2020.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems- Volume 2*, pages 3111–3119, 2013.
- Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6(1):1–10, 2016.
- SPFGH Moen and Tapio Salakoski2 Sophia Ananiadou. Distributional semantics resources for biomedical text processing. *Proceedings of LBM*, pages 39–44, 2013.
- Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. Deepr : a convolutional net for medical records. *IEEE journal of biomedical and health informatics*, 21(1):22–30, 2016.
- Junwoo Park, Youngwoo Cho, Haneol Lee, Jaegul Choo, and Edward Choi. Knowledge graph-based question answering with electronic health records. In *Machine Learning for Healthcare Conference*, pages 36–53. PMLR, 2021.

- Yifan Peng, Shankai Yan, and Zhiyong Lu. Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. *BioNLP 2019*, page 58, 2019.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, 2019.
- Huan Song, Deepta Rajan, Jayaraman J Thiagarajan, and Andreas Spanias. Attend and diagnose: Clinical time series analysis using attention models. In *Thirty-second AAAI conference on artificial intelligence*, 2018.
- Lihong Song, Chin Wang Cheong, Kejing Yin, William K Cheung, Benjamin CM Fung, and Jonathan Poon. Medical concept embedding with multiple ontological representations. In *IJCAI*, pages 4613–4619, 2019.
- Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2019.
- Harini Suresh, Nathan Hunt, Alistair Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Clinical intervention prediction and understanding with deep neural networks. In *Machine Learning for Healthcare Conference*, pages 322–337. PMLR, 2017.
- Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5103–5114, 2019.
- Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1499–1509, 2015.
- Truyen Tran, Tu Dinh Nguyen, Dinh Phung, and Svetha Venkatesh. Learning vector representation of medical objects via emr-driven nonnegative restricted boltzmann machines (enrbm). *Journal of biomedical informatics*, 54:96–105, 2015.
- Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: On the importance of pre-training compact models. *arxiv preprint arXiv:1908.08962*, 2019.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.
- Stephen H Walsh. The clinician’s perspective on electronic health records and how they can affect patient care. *Bmj*, 328(7449):1184–1187, 2004.
- Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph and text jointly embedding. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1591–1601, 2014.
- Han Xiao, Minlie Huang, Lian Meng, and Xiaoyan Zhu. Ssp: semantic space projection for knowledge graph embedding with text descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.
- Jiacheng Xu, Xipeng Qiu, Kan Chen, and Xuanjing Huang. Knowledge graph representation with jointly structural and textual encoding. In *IJCAI*, 2017.

Changchang Yin, Rongjian Zhao, Buyue Qian, Xin Lv, and Ping Zhang. Domain knowledge guided deep learning with electronic health records. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 738–747. IEEE, 2019.

Muhan Zhang, Christopher R King, Michael Avidan, and Yixin Chen. Hierarchical attention propagation for healthcare representation learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 249–256, 2020.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, 2019.

Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 267–272, 2015.

Appendix A. Further discussion

A.1. Abstract nodes & literal nodes

Unlike literal nodes whose vocabulary size is typically bounded (there is a finite number of unique medical codes), the vocabulary size of abstract nodes is, in theory, unbounded since new visits or new medication orders occur every day. Also, abstract nodes do not mean anything in themselves but are simply an aggregation of information from their children nodes. Therefore, to reduce the node space complexity and encourage efficient representation learning of the EHR graph, we replace each abstract node with a placeholder node. Specifically, admission, diagnosis, procedure, and prescription nodes are replaced with [ADM], [DX], [PX], and [RX], respectively.

Appendix B. Details of EHR Graph

B.1. Table to Knowledge Graph Conversion

Knowledge Graphs (KG) are a set of triples that describe the relationship between two entities or between an entity and a literal value (Chakraborty et al., 2021). To extract triples from the tables of MIMIC-III, (Park et al., 2021) defined an algorithm to map each column to either an entity, a literal, or a relation.

By following the approach (Park et al., 2021), we first choose five tables from MIMIC-III: diagnoses, procedures, prescriptions, diagnosis descriptions, and procedure descriptions. We then merge the procedures and their descriptions table into a single procedure table as well as a diagnoses table. The primary or foreign key columns of a table are defined as entities, and non-key columns (i.e., property columns) are defined as literals. For example, in the prescriptions table of Figure 4, the values of ADM_ID are defined as entities that can have child nodes. The values of DRUG, FORMULARY_DRUG_CD, and DOSE are defined as literals that cannot have child nodes. The column names of the non-key columns become relations, e.g. /diagnoses_long_title links between ICD9_CODE/4513 and “Other endoscopy of small intestine”. Lastly, the relation between a primary key and a foreign key is derived from the table name (e.g., /procedures links between ADM_ID/198041 and PX_ID/186630).

B.2. Relations in EHR graph

For Dx+Px, relation types are /diagnoses_icd9_code , /diagnoses , /procedures , /procedures_icd9_code , /di-

agnoses_long_title , /procedures_long_title . For Rx, relation types are /icustay_id , /prescriptions , /drug_type , /drug , /route , /formulary_drug_cd , /drug_dose .

Appendix C. Experiment details

C.1. Pretrain

Data preparation For the graph modality, we convert five MIMIC-III tables into the EHR graph as in B. For the text modality, we extract clinical notes from noteevents table in MIMIC-III. Among various kinds of notes, our target is the discharge summary, so we divide the text belonging to the discharge summary based on section headers and extract each section separately. The section headers corresponding to diagnoses, procedures, and prescriptions are discharge diagnosis, major surgical or invasive procedure, and discharge medication, respectively. The complete dataset is constructed by coupling a subgraph and text which belongs to the same admission id. We drop the admission that has less than five words in the text modality to guarantee the existence of text information. For Rx, we filter the subgraph which has more than 768 nodes including the admission node. A limitation of existing Transformer models and their derivatives is that the full self-attention mechanism has computational and memory requirements that are quadratic with the input sequence length. To fit our model into RTX 3090 with a reasonable batch size (at least 4), the maximum possible sequence length was near 768. This design choice truncates 22% of the samples in the dataset. We tried to increase the sequence length from 768 to 1024, and this change only restores the 5% of samples while the model does not fit in GPU.

As shown in Section 4.1 and Table 1, we merge diagnosis (Dx), procedures (Px) into one dataset (Dx+Px) without any modifications.

Implementation details We construct the graph vocabulary with unique abstract, literal nodes and three special tokens for a graph modality ([SUM], [MASK]_G, [PAD]_G). We use the same vocabulary which BERT used for the text. For the model’s input, the graph nodes are first listed in breadth first search (BFS) order followed by tokenized text subwords. We pad each modality inputs to the maximum length and add special tokens.

The two essential pre-training tasks, masked language modeling (MLM) and masked literal prediction

DX/DIAGNOSES					
DX_ID	ADM_ID	ICD9_CODE	...		
630974	198041	9351	...		
630975	198041	042	...		
DX/D_ICD_DIAGNOSES					
ICD_ID	ICD9_CODE	SHORT_TITLE	LONG_TITLE		
513	042	Human immuno virus dis	Human immunodeficiency virus [HIV] disease		
12338	9351	Foreign body esophagus	Foreign body in esophagus		
PX/PROCEDURES					
PX_ID	ADM_ID	ICD9_CODE	...		
186630	198041	4513	...		
186631	198041	9802	...		
PX/D_ICD_PROCEDURES					
ICD_ID	ICD9_CODE	SHORT_TITLE	LONG_TITLE		
1399	4513	Sm bowel endoscopy NEC	Other endoscopy of small intestine		
3795	9802	Remov intralum esoph FB	Removal of intraluminal foreign body...		
RX/PRESCRIPTIONS					
RX_ID	ADM_ID	DRUG	FORMULARY_DRUG_CD	DOSE	...
1310323	198041	Midazolam	MIDA2I	2mg	...
1310324	198041	Fentanyl Citrate	FENT2I	50mcg	...

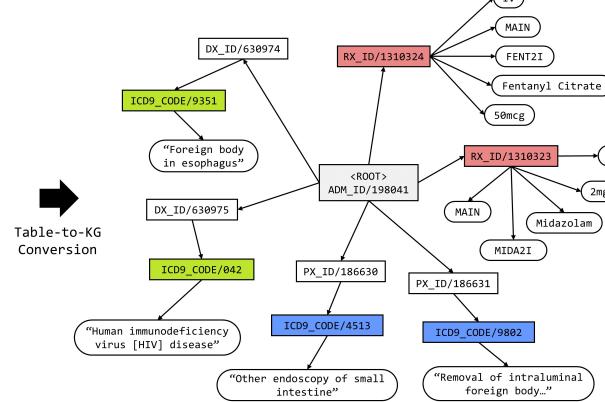


Figure 4: Following (Park et al., 2021), five tables in MIMIC-III (*i.e.*, diagnoses, d_icd_diagnoses, procedures, d_icd_procedures, prescriptions) are converted to a knowledge graph. Rectangles and ellipses in the graph indicate entities and literal values, respectively. We omit relations from the figure for simplicity.

(MLP), follow the standard masking strategy in BERT: randomly masking out 15% of the tokens in the input, and pass the masked sequence through the model and predict the masked tokens. In MLP, only literal nodes are treated as masking targets, and both MLM and MLP use the cross entropy loss over the masked tokens as an objective function. For relation classification (RC), we provide the position of two nodes and the type of relation between them as an input, and the outputs of two nodes are concatenated and pass through the multi-class classifier. For cross-modal alignment prediction (AP), we follow the same setting in Section 3.2. Note that RC and AP also use the cross entropy loss to optimize the model during pre-training.

Model hyperparameter settings We set the hidden dimension of MedGTX and all baselines to 128. The number of blocks in each single modality encoder is 2, and 4 for the cross-modal encoder. The number of attention heads is always 4.

C.2. Cross-modal retrieval

Data preparation We use the same data and splits in Section C.1. We sample the negative pair inside the batch where the graph or text belongs to another admission.

Training & Evaluation For fine-tuning, we concatenate [SUM], [CLS] together and feed them into

two layers of feed-forward network with a non-linear activation function (*tanh*) between those layers. The classifier predicts the cross-modal alignment score of the graph-text pair by taking softmax on the output.

During the inference, we consequently sample the graph (text) from the test set, and measure the cross-modal alignment score with the entire text (graph) in the test set. We calculate the rank based on cross-modal alignment score where a higher cross-modal alignment score gets a higher position in rank.

C.3. Temporal prediction

Data preparation We select patient cohorts starting from the same data and splits in Section C.1. For the readmission prediction task, each dataset is divided into two groups based on whether readmitted (1) or not (0) within 30 days after discharge. Similarly, for the mortality prediction task, the target label depends on whether a patient is either dead within 30 days (1) or not (0). For Dx+Px, the final cohorts for hospital readmission and mortality prediction contain 6,948 and 26,577 training samples with 2,072 and 1,439 positive labels, respectively. For Rx, for the same downstream tasks, the final cohorts contain 5,808 and 23,412 training samples with 1,597 positive labels and 1,108 positive labels, respectively.

Training & Evaluation For training, we first pass the inputs to the model and then concatenate the output vectors which correspond to [SUM], [CLS]

Table 10: Hyperparameters for each task.

Hyperparameters	Pre-training	Cross-modal retrieval	Temporal prediction	Error detection	Clinical note generation
Learning rate	1×10^{-4}	1×10^{-5}	1×10^{-4}	1×10^{-5}	3×10^{-5}
Training epochs	40	20	20	20	30
Number of negative sample	1	1	1	-	-
MLM probability	0.15	-	-	-	0.15
Corruption probability	-	-	-	0.25	-
p (Top- p sampling)	-	-	-	-	0.9
Dropout probability			0.1		
Batch size		16 (Dx+Px) / 8 (Rx)			
Random seed		1, 12, 123, 1234, 42			
Maximum length of text			512		

together and feed them into two layers of feed-forward network with a non-linear activation function (*tanh*) between those layers. For the temporal prediction task, we use binary classifier outputs via sigmoid operations. We optimize the model by minimizing binary cross entropy loss over predicted and ground truth labels.

C.4. Error detection

Data preparation We randomly replace the literals in the graph with a probability of 0.25. For Rx graphs, we alter five types of literal nodes (*i.e.* drug name, drug code, route, formulary drug cd, and dosage). For Dx+Px graphs, we alter two types of literal nodes (*i.e.* diagnosis name, procedure name). We choose the alternative that belongs to the same type as the original node.

Training & Evaluation We add a one-versus-all (OVA) binary classification head on the top of MedGTX. The OVA classifier receives a contextualized embedding of each node as input and predicts whether each node in the EHR graph is altered (1) or not (0). To be specific, the OVA classifier consists of a simple linear layer and optimizes the model by using binary cross-entropy loss. For evaluation metric, we use F1 score to report the model performance.

C.5. Clinical note generation

Data preparation We use the same data and splits in Section C.1. To directly leverage source database (*i.e.* MIMIC-III) information for the generation task, we do not use specialized meta-tags or pre-processing techniques.

Training & Evaluation Following the fine-tuning strategy in UniLM (Dong et al., 2019), we fine-tune the model on masked language modeling task, but

the attention mask is specialized for a sequence-to-sequence (S2S) task. This S2S attention mask adds the causality constraint to the text modality while the graph nodes can attend any other tokens (include itself) in a graph. Note that it’s impossible to maintain uni-directionality in a text modality with a cross attention mechanism because the bidirectional information of the text flows through the previous cross-modal attention layers. Therefore we cut off one of the cross-modal attention (*i.e.* CrossAttn $_{G \rightarrow T}$), and use a causal attention mask for the text modality. We also masked the [SEP] token with a masking probability of 0.5 to encourage the model to emit [SEP] token which terminates the generation process.

During the evaluation, we first pack the whole graph inputs and [CLS] into a sequence. We append [MASK] token to the last of a sequence and predict the masked token. We run this procedure in an auto-regressive and greedy manner to decode a sequence until the [SEP] token comes out. For the Dx+Px dataset, we keep decoding until the second [SEP] token is emitted. Also, we change section embeddings after the model emits the first [SEP] token. Since we choose the optimal decoding strategy over all models for each dataset, we adopt top- p sampling strategy for Rx.

Appendix D. Cross attention visualization

In this section, we visualize the additional samples which belong to one of three types of connection between two modalities: 1) same surface form (in Figure 5); 2) different surface forms, but represent the same medical concept (in Figure 6, 7); 3) different medical concepts, but capture a biomedical correlation between concepts (in Figure 8).

Appendix E. Important hyperparameters

We specify hyperparameter settings including pre-training and all downstream tasks in Table 10.

Appendix F. Experiment environments

We train our models on NVIDIA GeForce RTX 3090 and RTX A6000. Also, We use CUDA 11.1, and PyTorch 1.9.0. for experiments.



Figure 5: **Same surface form**; The node “*urinary tract infection, site not specified*” (graph) highlights {“*urinary*”, “*tract*”, “*infection*”} (text) in the cross attention map. Although the MedGTX does not use information about the surface form of words, it can be seen that attention weights are strongly applied to words with the same meaning.

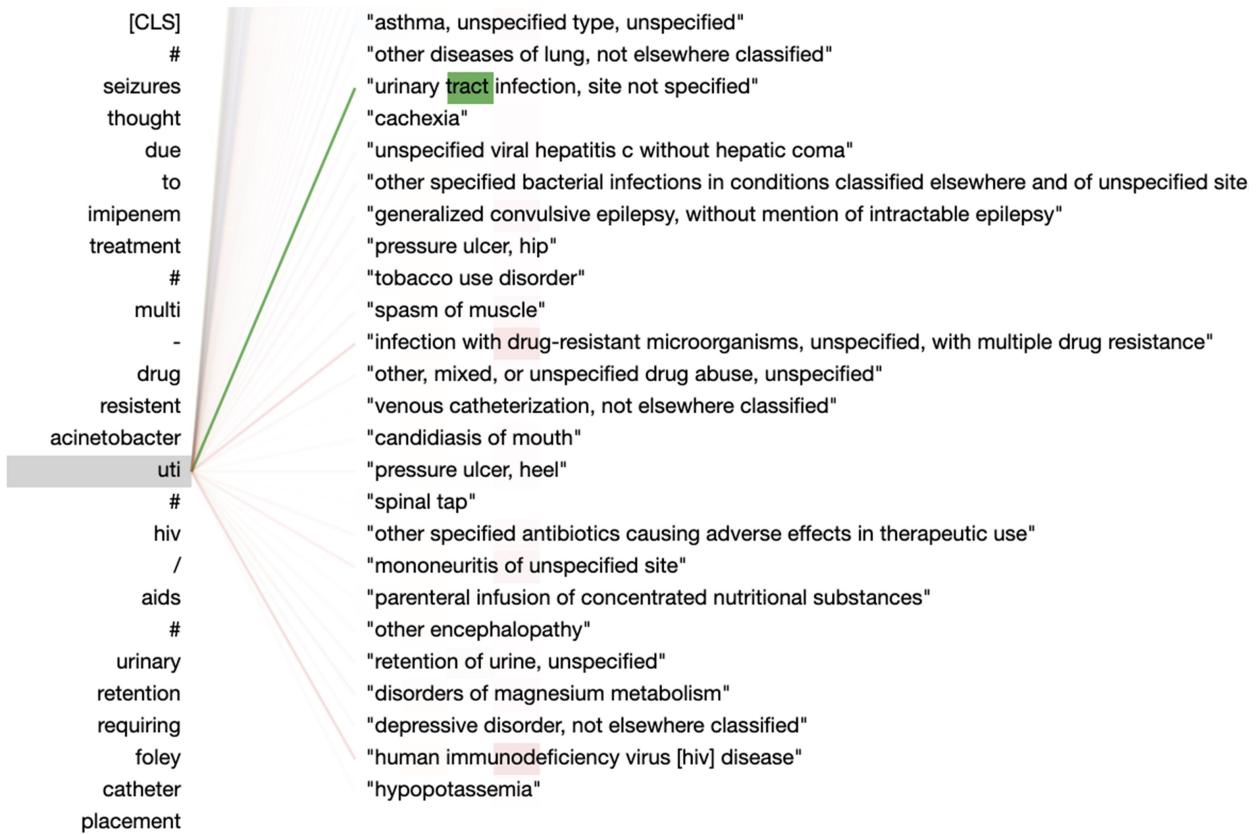


Figure 6: Different surface forms, but represent the same medical concept; The token “*uti*” (text) highlights “*urinary tract infection, site not specified*” (graph) in the cross attention map. “*uti*” is an abbreviation of “*urinary tract infection*”. They have different surface forms but have the same meaning.

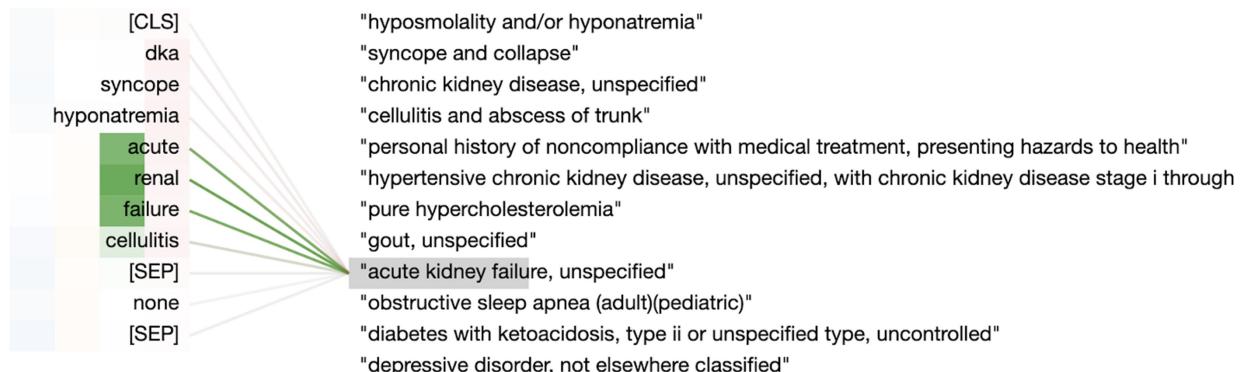


Figure 7: Different surface forms, but represent the same medical concept; The node “*acute kidney failure, unspecified*” (graph) highlights {“acute”, “renal”, “failure”} (text) in the cross attention map. “*renal*” differs from “*kidney*” in the appearance, but they have the same meaning.

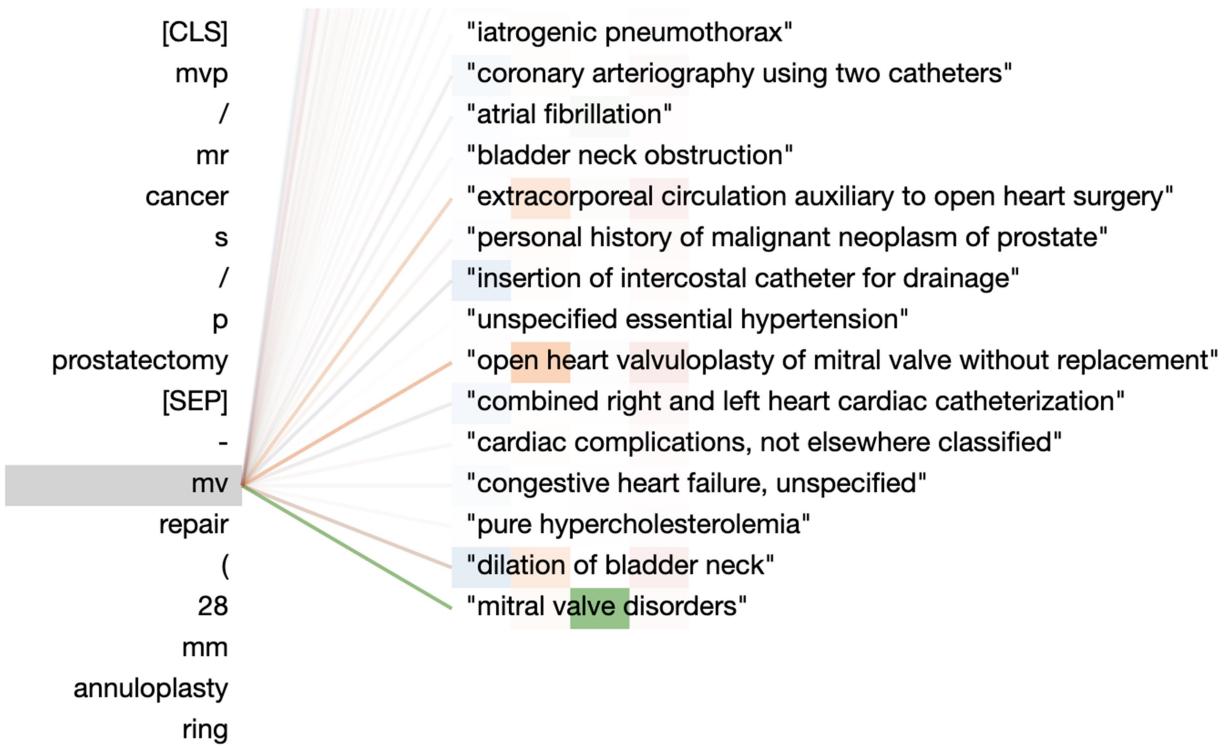


Figure 8: **Different medical concepts, but capture a biomedical correlation between concepts;** The token “*mv*” (text) highlights “*mitral valve disorders*” (graph) in the cross attention map. “*mv*” is an abbreviation for “*mitral valve*”. They have different surface forms but have the same meaning. Furthermore, “*mv*” also strongly attends on “*open heart valvuloplasty of mitral valve without replacement*” and “*extracorporeal circulation auxiliary to open heart surgery*” nodes. They have different medical concepts, but it captures a biomedical correlation between concepts.