

# Knowledge-driven Encode, Retrieve, Paraphrase for Medical Image Report Generation

Christy Y. Li<sup>\*1</sup>, Xiaodan Liang<sup>†2</sup>, Zhiting Hu<sup>2</sup>, Eric P. Xing<sup>3</sup>

<sup>1</sup>Duke University, <sup>2</sup>Carnegie Mellon University, <sup>3</sup>Petuum, Inc  
yl558@duke.edu, {xiaodan1,zhitingh}@cs.cmu.edu, eric.xing@petuum.com.

## Abstract

Generating long and semantic-coherent reports to describe medical images poses great challenges towards bridging visual and linguistic modalities, incorporating medical domain knowledge, and generating realistic and accurate descriptions. We propose a novel *Knowledge-driven Encode, Retrieve, Paraphrase* (KERP) approach which reconciles traditional knowledge- and retrieval-based methods with modern learning-based methods for accurate and robust medical report generation. Specifically, KERP decomposes medical report generation into explicit medical abnormality graph learning and subsequent natural language modeling. KERP first employs an *Encode* module that transforms visual features into a structured abnormality graph by incorporating prior medical knowledge; then a *Retrieve* module that retrieves text templates based on the detected abnormalities; and lastly, a *Paraphrase* module that rewrites the templates according to specific cases. The core of KERP is a proposed generic implementation unit—Graph Transformer (GTR) that dynamically transforms high-level semantics between graph-structured data of multiple domains such as knowledge graphs, images and sequences. Experiments show that the proposed approach generates structured and robust reports supported with accurate abnormality description and explainable attentive regions, achieving the state-of-the-art results on two medical report benchmarks, with the best medical abnormality and disease classification accuracy and improved human evaluation performance.

## Introduction

Beyond the traditional image captioning task (Xu et al. 2015; Karpathy and Fei-Fei 2015; Rennie et al. 2017) that produces single-sentence descriptions, generating long and semantic-coherent stories or reports to describe visual contents (e.g., images, videos) has recently attracted increasing research interests (Liang et al. 2017; Huang et al. 2016; Krause et al. 2017), and is posed as a more challenging and realistic goal towards bridging visual patterns with human linguistic descriptions. Particularly, an outstanding challenge in modeling long narrative from visual content is

to balance between knowledge discovery and language modeling (Karpathy and Fei-Fei 2015). Current visual text generation approaches tend to generate plausible sentences that look natural by the language model but poor at finding visual groundings. Although some approaches have been proposed to alleviate this problem (Lu et al. 2018; Anderson et al. 2018; Liang et al. 2017), most of them ignore the internal knowledge structure of the task at hand. However, most real-world data and problems exhibit complex and dynamic structures such as intrinsic relations among discrete entities under nature’s law (Taskar, Guestrin, and Koller 2004; Hu et al. 2016; Strubell et al. 2018). Knowledge graph, as one of the most powerful representations of dynamic graph-structured knowledge (Mitchell et al. 2018; Bizer, Heath, and Berners-Lee 2011), complements the learning-based approaches by explicitly modeling the domain-specific knowledge structure and relational inductive bias. Knowledge graph also allows incorporating priors, which is proven useful for tasks where universal knowledge is desired or certain constraints have to be met (Battaglia et al. 2017; Liang, Hu, and Xing 2018; Hu et al. 2018; X. Wang 2018).

As an emerging task of long text generation of practical use, *medical image report generation* (Li et al. 2018; Jing, Xie, and Xing 2018) must satisfy more critical protocols and ensure the correctness of medical terminology usage. As shown in Figure 1, a medical report consists of a finding section describing medical observations in details of both normal and abnormal features, an impression or conclusion sentence indicating the most prominent medical observation, and peripheral sections such as patients information and indications. Among these sections, the finding section is considered as the most important component and is expected to 1) cover contents of key relevant aspects such as heart size, lung opacity, and bone structure; 2) correctly detect any abnormalities and support with details such as the location and shape of the abnormality; 3) describe potential diseases such as effusion, pneumothorax and consolidation.

It is often observed that, to write a medical image report, radiologists first check a patient’s images for abnormal findings, then write reports by following certain patterns and templates, and adjusting statements in the templates for each individual case when necessary (Hong and Kahn 2013). To mimic this procedure, we propose to formulate medical report writing as a knowledge-driven encode, retrieve, para-

<sup>\*</sup>This work was done when Christy Y. Li was at Petuum, Inc.

<sup>†</sup>Corresponding author.

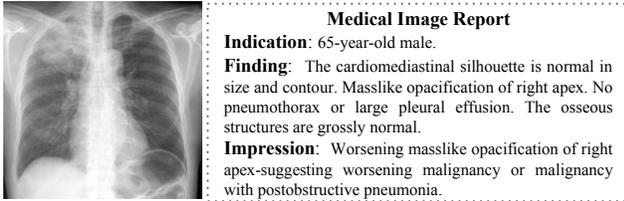


Figure 1: Example of medical image report

phrase (KERP) process. In particular, KERP first invokes an *Encode* module to transform visual features of medical images into an abnormality graph where each node represents a possible clinical abnormality designed by prior medical knowledge, and the features of which depict semantics of the abnormality that can be observed from the visual input (e.g., normal or abnormal, size, location). The correlation of abnormality nodes is further encoded as edge weights of the abnormality graph so that the relations among different abnormal findings are considered when making a clinical diagnostic decision. Then KERP retrieves a sequence of templates according to the detected abnormalities via a *Retrieve* module. The words of the retrieved templates are further expanded and paraphrased into a report by a *Paraphrase* module which enriches the templates with details and corrects false information if any.

As most real-world data (e.g., images, sequence of text tokens, knowledge graphs, convolutional feature maps) can be represented as graphs, we further propose a novel and generic implementation unit—Graph Transformer (GTR) which dynamically transforms among multi-domain graph-structured data. We further equip GTR with attention mechanism for learning robust graph structure, as well as incorporating prior knowledge structure. By invoking GTR, KERP can transform robustly from visual features to an abnormality graph (with the *Encode* module), then to sequences of templates (with the *Retrieve* module), and lastly to sequences of words (with the *Paraphrase* module).

We conduct extensive experiments on two medical image report dataset (Demner-Fushman et al. 2015). Our KERP achieves the state-of-the-art performance on both datasets under both automatic evaluation metrics and human evaluation. KERP also achieves best performance on abnormality classification. Experiments show that KERP not only generates structured and robust reports supported with accurate abnormality prediction, but also produces explainable attentive regions which is crucial for interpretative diagnosis.

## Related Work

**Medical report generation.** Machine learning for healthcare has been widely recognized in both academia and industry as an area of high impact and potential. Automatic generation of medical image reports, as one of the key applications in the field, is gaining increasing research interest (Li et al. 2018; Wang et al. 2018b; Jing, Xie, and Xing 2018). The task differs from other tasks such as summarization where summaries tend to be more diverse without clear

templates or internal knowledge structure; and image captioning where usually a single sentence is desired.

**Graph neural networks.** Graph neural networks (GNN) have gained increasing research interests (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017; Monti, Bronstein, and Bresson 2017). However, most existing methods learn to encode the input feature into higher-level feature through selective attention over the object itself (Wang et al. 2018a; Parmar et al. 2018; Velickovic et al. 2018), while our method works on multiple graphs, and models not only the data structure within the same graph but also the transformation rules among different graphs.

**Hybrid retrieval-generation approach.** Combining traditional retrieval-based and modern generation-based methods for (long) text generation (Li et al. 2018; Guu et al. 2018; Ziqiang Cao and Wei 2018; Hu et al. 2017) has gained increasing research interests. Our work differs from previous work in that: 1) we develop an encoding procedure that explicitly learns the graph structure of medical abnormalities; 2) the *retrieve* and *rerank* is formulated as one joint, comprehensive process and implemented via a novel and generic unit—Graph Transformer.

## Graph Transformer (GTR)

We start by describing Graph Transformer (GTR) which transforms a graph into another graph for encoding features into higher-level semantics within the same graph type, or translating features of one graph (e.g., knowledge graph) into another one (e.g., sequence of words). First, we represent a graph as  $G = (V, E)$ . Here  $V = \{\mathbf{v}_i\}_{i=1:N}$  is a set of nodes where each  $\mathbf{v}_i \in \mathbb{R}^d$  represents a node’s feature of dimension  $d$ , and  $N$  is the number of nodes in the graph.  $E = \{e_{i,j}\}_{i,j=[1,N]}$  is a set of edges between any possible pair of nodes. Here we study the setting where each edge is associated with a scalar value indicating closeness of nodes, while it is straightforward to extend the formalism to other cases where edges are associated with non-scalar values such as vectors.

GTR takes a graph  $G = (V, E)$  as input, and outputs another graph  $G' = (V', E')$ . Note that  $G$  and  $G'$  are two different graphs and can have different structures and characteristics (e.g.,  $N \neq N'$ ,  $d \neq d'$ , and  $e_{i,j} \neq e'_{i,j}$ ). This differs from many previous methods (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017; Velickovic et al. 2018) which are restricted to the same graph structures. For both source and target graph, the set of nodes  $V$  and  $V'$  has to be given in prior (e.g., the vocabulary size if the considered graph is sequences, abnormality nodes if the considered graph is an abnormality graph). We consider two scenarios for the edges among graph nodes: 1) the edges are provided in prior, and denoted as  $e_{s_i,t_j}$  where  $s_i$  is the  $i_{th}$  node of source graph and  $t_j$  is the  $j_{th}$  node of target graph; 2) the edges are not provided, and thus source and target nodes are represented as fully connected with uniform weights. We assume  $e_{s_i,t_j}$  as normalized, to avoid notation of averaging.

There are two types of message passing in GTR: from source graph to target graph (inter-graph message passing),

and message passing within the same graph (intra-graph message passing).

**Inter-graph message passing** To learn the source graph’s knowledge, the features of source nodes are transformed and passed to target nodes with their corresponding edge weights. The formulation can be written as:

$$\mathbf{v}'_j = \mathbf{v}'_j + \sigma\left(\sum_{i=1}^N e_{s_i,t_j} \mathbf{W}_s \mathbf{v}_i\right) \quad (1)$$

where  $\sigma$  is a nonlinear activation, and  $\mathbf{W}_s$  is a projection matrix of size  $d' \times d$ .

Considering that the edge information between source and target graphs may not be available in many cases (e.g., translating a sequence of words into another sequence of words), we propose to learn edge weights automatically by an attention mechanism (Vaswani et al. 2017). In this way, target node update is enabled to consider the varying importance of the source nodes. Specifically,

$$\hat{e}_{s_i,t_j} = \text{Attention}(\mathbf{W}_s^a \mathbf{v}_i, \mathbf{W}_t^a \mathbf{v}'_j) \quad (2)$$

where  $\hat{e}_{s_i,t_j}$  is the attention weight of edge from source node  $i$  to target node  $j$ ;  $\mathbf{W}_s^a$  and  $\mathbf{W}_t^a$  are weights in attention mechanism to project nodes features of source graph and target graph to a common space of dimension  $q$  respectively; and  $\text{Attention}: \mathbb{R}^q \rightarrow \mathbb{R}$  is the attention mechanism that transforms the two projected features  $\mathbf{W}_s^a \mathbf{v}_i, \mathbf{W}_t^a \mathbf{v}'_j \in \mathbb{R}^q$  to a scalar  $\hat{e}_{s_i,t_j}$  as the edge’s attention weight. In our experiments,  $\text{Attention}$  is parameterized as a scaled dot-product operation with multi-head attention (Vaswani et al. 2017).

The attention weights are normalized over all source nodes for each target node, denoting the relative importance of each source node to a target node among all source nodes. The formulation can be written as:

$$\hat{e}_{s_i,t_j} = \text{softmax}_{s_i}(\hat{e}_{s_i,t_j}) = \frac{\exp(\hat{e}_{s_i,t_j})}{\sum_{k=1}^N \exp(\hat{e}_{s_k,t_j})} \quad (3)$$

Once obtained, the normalized attention coefficients are combined with prior edge weights to pass features of connected source nodes to target nodes. The combined features are served as the target node’s updated features with source graph knowledge encoded. We adopt weighted sum of the learned attention edge weights and prior edge weights as final edge weights. Other methods such as multiplication of learned and prior edge weights followed by softmax also works. However, in our experiments, we observed that the first method performs better and avoids under-fitting. The formulation can be written as:

$$\tilde{e}_{s_i,t_j} = \lambda e_{s_i,t_j} + (1 - \lambda) \hat{e}_{s_i,t_j} \quad (4)$$

where  $\lambda$  is a user-defined weight controlling importance of prior edges and learned edges. If  $\lambda$  is set to 1, the edges between source graph and target graph are fixed, and no attention mechanism is required. The formulation is then the same as Equation 1. If  $\lambda$  is set to 0, the edges between source graph and target graph are completely learned by the model. With the updated weight, one can obtain updated target nodes features via Equation 1.

**Intra-graph message passing** Intra-graph message passing aims at modeling the correlation among nodes of the same graph, and fusing features according to the closeness

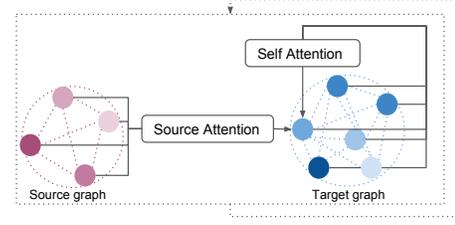


Figure 2: Architecture of *Graph Transformer*. GTR evolves a target graph by recurrently performing *Source Attention* on a source graph and *Self Attention* on itself. The darkness of color of each graph node indicates the degree of attention the target node pays to.

between them. Specifically, a target node is updated by combining features of neighboring nodes and itself. The formulation can be written as:

$$\mathbf{v}'_j = \mathbf{v}'_j + \sigma\left(\sum_{i=1}^{N'} \tilde{e}_{i,j} \mathbf{W}_t \mathbf{v}'_i\right) \quad (5)$$

where  $\mathbf{W}_t$  is weight to project features of target nodes from dimension  $d$  to output dimension. To learn the edge weights through attention mechanism, one can directly apply Equations 1-4 by changing source and target nodes notation to be of the same graph.

**GTR as a module** As shown in Figure 2, we formulate GTR as a module denoted as *GTR* by first concatenating intra-graph message passing and inter-graph message passing into one step (that is, first conduct message passing within target graph, then conducting message passing from one or multiple source graphs), then stacking multiple such steps into one module in order to progressively convert target graph features into high-level semantics.

**GTR for multiple domains** Most real-world data types (e.g., images, sequences, graphs) can be formulated as graph-structured. For example, a 2-dimensional image can be formulated as a graph whose nodes are pixels of the image where every node is connected with its neighboring pixel nodes; and a sequence of words can be formulated as a graph whose nodes are the individual words where edges among nodes are the consecutive relation among words. If global context of the data is considered, which is commonly adopted in attention mechanism (Vaswani et al. 2017), the graph nodes are then fully-connected. In the following, we describe the variants of *GTR* for different data domains by first formulating data as graph-structured, and then perform GTR operations on it. In particular, we define  $GTR_{i2g}$  as the variant of *GTR* for transforming image features into graph’s features;  $GTR_{g2g}$  the variant of *GTR* for transforming from a graph to another graph;  $GTR_{g2s}$  the variant of *GTR* for graph input and sequence output; and  $GTR_{gs2s}$  the variant of *GTR* for graph and sequence input and sequence output. We use the variants of *GTR* as building blocks of KERP for medical report generation, which is described in section.

**GTR for sequential input/output.** To apply GTR for sequential input or output (e.g., a sequence of words, a sequence of retrieved items), we employ *positional encod-*

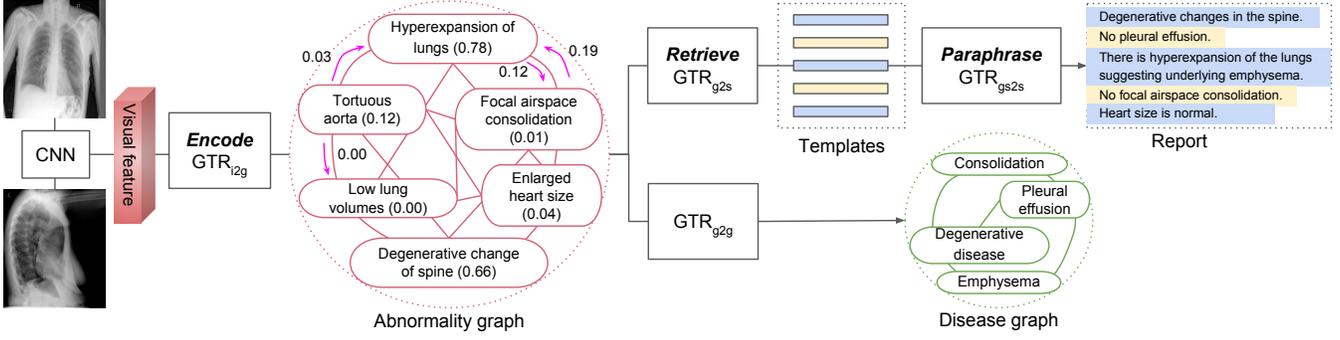


Figure 3: Architecture of KERP. Image features are first extracted from a CNN, and further encoded as an abnormality graph via *Encode GTR<sub>i2g</sub>*. *Retrieve GTR<sub>g2s</sub>* decodes the abnormality graph as a template sequence, the words of which are then retrieved and paraphrased by *Paraphrase GTR<sub>gs2s</sub>* as the generated report. Simultaneously, a *GTR<sub>g2g</sub>* decodes the abnormality graph as a disease graph, and predicts disease categories via extra classification layers. In the abnormality graph, values inside parentheses are probabilities of the corresponding nodes predicted by extra classification layers taking latent semantic features of nodes as input. Values along the directed arrows indicate attention scores of source nodes on target nodes.

ing (Vaswani et al. 2017) to GTR so as to add relative and absolute position information to the input or output sequence. Specifically, we use sine and cosine functions of different frequencies:

$$PE_{pos,2i} = \sin(pos/10000^{2i/d}) \quad (6)$$

$$PE_{pos,2i+1} = \cos(pos/10000^{2i/d}) \quad (7)$$

where  $pos$  is the position and  $i$  is the dimension. If both input and output are sequences, GTR is close to a *Transformer* (Vaswani et al. 2017) with prior edge weights.

**GTR for image input.** We denote visual features of an image as  $I \in R^{D,W,H}$  where  $D$  is the dimension of latent features,  $W$  and  $H$  is width and height. To apply GTR for image input, we first reshape visual features by flattening the 2-dimension into 1-dimension  $R^{W \times H, D}$ . Then each pixel is treated as graph node whose features are used as source graph features.

**GTR for multiple input graphs.** For the cases where a target graph wants to learn from more than one source graphs, we extend *GTR* to take into account multiple input source by replacing the single inter-graph message passing in each stacked layer of *GTR* into multiple concatenated inter-graph message passing.

## Knowledge-driven Encode, Retrieve, Paraphrase (KERP)

It is observed that, to write a medical image report, radiologists first check a patient’s images for abnormal findings, then write reports by following certain patterns and templates, and adjusting statements in the templates for each individual case when necessary (Hong and Kahn 2013). To mimic this procedure, we propose to formulate medical report writing as a process of encoding, retrieval and paraphrasing. In particular, we first compile an off-the-shelf abnormality graph that contains large range of abnormal findings. We consider frequent abnormalities stem from thoracic

organs as nodes in the abnormality graph. For example, “disappearance of costophrenic angle”, “low lung volumes”, and “blunted costophrenic angle”. These abnormalities compose a major part of medical reports, and their detection quality would greatly impact the accuracy of the generated reports. Please see Appendix for detailed definition and examples. We also compile a template database that consists of a set of frequent sentences that cover descriptions of different abnormalities in the abnormality graph. For example, “there is hyperexpansion of lungs and flattening of the diaphragm consistent with copd” is a template for describing “hyperexpansion of lungs”, “flattening of diaphragm” and “copd”.

Then we design separate modules for the purpose of encoding visual features as an abnormality graph, retrieving templates based on the detected abnormalities, and rewriting templates according to case-specific scenario. As described in Figure 3, a set of images are first fed into a CNN for extracting visual features which are then transformed into an abnormality graph via *Encode GTR<sub>i2g</sub>*. *Retrieve GTR<sub>g2s</sub>* decodes the abnormality graph as a template sequence, the words of which are then retrieved and paraphrased by *Paraphrase GTR<sub>gs2s</sub>* as the generated report.

In addition, we design a disease graph containing common *thorax* diseases (e.g., nodule, pneumonia and emphysema) which are commonly concluded from single or combined condition of abnormalities. For example, atelectasis may be concluded if “interval development of bandlike opacity in the left lung base” is present; consolidation and atelectasis may exist if there is “streaky and patchy bibasilar opacities”, and “triangular density projected over the heart” without “typical findings of pleural effusion or pulmonary edema”. In parallel to generating reports in the proposed model architecture, a *GTR<sub>g2g</sub>* is employed to transform the abnormality graph to a disease graph in order to predict common *thorax* diseases (right lower column of Figure 3) which can be useful as concluding information for medical reports.

## Encode: visual feature to knowledge graph

The *Encode* module aims at encoding visual features as an abnormality graph. Assume an input image is encoded with a deep neural network into feature  $\mathbf{X} \in R^{W \times H \times d_X}$  where  $W$ ,  $H$  and  $d_X$  are width, height, and feature dimension, respectively. An abnormality graph is represented as a set of nodes of size  $N$  with initialized features. The latent features of each node can be used to predict occurrence of the abnormality via an additional classification layer. By applying the variant of GTR for image input and graph output, denoted as  $GTR_{i2g}$ , the updated node features can be written as:

$$\mathbf{h}_u = GTR_{i2g}(\mathbf{X}) \quad (8)$$

$$\mathbf{u} = \text{sigmoid}(\mathbf{W}_u \mathbf{h}_u) \quad (9)$$

where  $GTR_{i2g}$  is the formulation of the variant of *GTR* for image input and graph output described on page 3, and  $\mathbf{W}_u$  is linear projection to transform latent feature  $u$  into 1-d probability.  $\mathbf{h}_u = (\mathbf{h}_{u_1}; \mathbf{h}_{u_2}; \dots; \mathbf{h}_{u_N}) \in R^{N \times d}$  is the set of latent features of nodes where  $d$  is feature dimension.  $\mathbf{u} = (u_1, u_2, \dots, u_N), y_i \in \{0, 1\}, i \in \{1, \dots, N\}$  denotes binary label for abnormality nodes.

## Retrieve: knowledge graph to template sequence

Once the knowledge graph of abnormality attributes is obtained, it can be used to guide the retrieval process to harvest templates. A sequence of templates is represented as  $\mathbf{t} = (t_1, t_2, \dots, t_{N_s})$  where  $N_s$  is the maximum length of template sequence (also maximum number of sentences) and  $t_i$  is index of templates where  $i \in \{1, \dots, N_s\}$ . By applying the variant of *GTR* for graph input and sequential output, denoted as  $GTR_{g2s}$ , the retrieved template sequence can be obtained as follows:

$$\mathbf{h}_t = GTR_{g2s}(\mathbf{h}_u) \quad (10)$$

$$\mathbf{t} = \arg \max \text{Softmax}(\mathbf{W}_t \mathbf{h}_t) \quad (11)$$

where  $GTR_{g2s}$  denotes the formulation of the variant of *GTR* for graph input and sequence output,  $\mathbf{W}_t$  is linear projection to transform latent feature to template embedding, and  $\arg \max \text{Softmax}$  is the operation that selects index of maximum value after *Softmax* function.

## Paraphrase: template sequence to report

*Paraphrase* serves for two purposes: 1) refine templates with enriched details and possibly new case-specific findings; 2) convert templates into more natural and dynamic expressions. The first purpose is achieved by modifying information in the templates that is not accurate for specific cases, and the second purpose is achieved by robust language modeling for the same content. These two goals supplement each other in order to generate accurate and robust reports.

Given the retrieved sequence of templates  $\mathbf{t} = (t_1, t_2, \dots, t_{N_s})$ , the rewriting step generates a report consisting of a sequence of sentences  $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N_s})$  by subsequently attending to template words and the encoded knowledge graph (described on page 4). Each sentence  $\mathbf{r}_i = (w_{i1}, w_{i2}, \dots, w_{iN_w}), i \in \{1, \dots, N_s\}$  consists of a sequence of  $N_w$  words.

$$\mathbf{h}_w = GTR_{gs2s}(\mathbf{h}_u, \mathbf{t}) \quad (12)$$

$$\mathbf{R} = \arg \max \text{Softmax}(\mathbf{W}_w f(\mathbf{h}_w)) \quad (13)$$

where  $GTR_{gs2s}$  denotes the formulation of the variant of *GTR* for graph and sequence input, and sequence output,  $f$  denotes the operation of reshaping  $\mathbf{h}_w$  from  $R^{N_s \times N_w \times d}$  to  $R^{N_s \times N_w \times d}$ ,  $\mathbf{W}_w$  is linear projection to transform latent feature into word embedding, and  $\arg \max \text{Softmax}$  selects index of maximum value after *Softmax* function.

## Disease classification

Abnormality attributes are frequently used for diagnosing diseases. Let  $\mathbf{z} = (z_1, z_2, \dots, z_L)$  be a one-hot representation of disease labels where  $L$  is the total number of disease labels, and  $z_i \in \{0, 1\}, i \in \{1, \dots, L\}$ . The multi-label disease classification can be written as:

$$\mathbf{h}_z = GTR_{g2g}(\mathbf{h}_u) \quad (14)$$

$$\mathbf{z} = \text{sigmoid}(\mathbf{W}_z \mathbf{h}_z) \quad (15)$$

where  $GTR_{g2g}$  denotes the formulation of the variant of *GTR* for graph input and graph output,  $\mathbf{W}_z$  is linear projection to transform disease nodes feature into 1-d probability.

## Learning

During paraphrasing, the retrieved templates  $\mathbf{t}$ , instead of latent feature  $\mathbf{h}_t$ , is used for rewriting. Sampling the templates of maximum predicted probability breaks the connectivity of differentiable back-propagation of the whole *encode-retrieve-paraphrase* pipeline. To alleviate this issue, we first train the *Paraphrase* with ground truth templates, and then with sampled templates generated by *Retrieval* module. This minimizes the negative effect of dis-connectivity and make better test-time performance by letting the model accommodate to its own generated templates. Besides, given that templates serve as starting points instead of ground truth for rewriting, the prediction of templates does not have to be highly accurate as the *Paraphrase* module needs to learn to paraphrase any suitable templates.

## Experiments & Results

**Dataset.** We conduct experiments on two medical image report datasets. First, **Indiana University Chest X-Ray Collection (IU X-Ray)** (Demner-Fushman et al. 2015) is a public dataset consisting of 7,470 chest x-ray images paired with their corresponding diagnostic reports. Each patient has 2 images (a frontal view and a lateral view) and a report which includes impression, finding and indication sections. We preprocess the reports by tokenizing and converting to lower-cases. We filter tokens by minimum frequency 3, which results in 1185 unique tokens covering over 99.0% word occurrences in the corpus. We collect 80 abnormalities and 87 templates for IU X-Ray dataset. **CX-CHR** is a private dataset of chest X-ray images with corresponding Chinese reports collected from a professional medical institution for health checking. The dataset consists of 35,609 patients and 45,598 images. Each patient has one or multiple chest x-ray images in different views (e.g., frontal and lateral), and a corresponding Chinese report. We select patients with no more than 2 images and obtain 33,236 patient samples in total which covers over 93% of the dataset. We preprocess reports by tokenizing and filtering tokens whose

Dataset	Model	CIDEr	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4	Hit (%)
IU X-Ray	CNN-RNN	0.294	0.307	0.216	0.124	0.087	0.066	-
	LRCN	0.285	0.307	0.223	0.128	0.089	0.068	-
	AdaAtt	0.296	0.308	0.220	0.127	0.089	0.069	-
	Att2in	0.297	0.307	0.224	0.129	0.089	0.068	-
	CoAtt*	0.277	<b>0.369</b>	0.455	0.288	0.205	0.154	24.100
	HRGR-Agent	<b>0.343</b>	0.322	0.438	0.298	0.208	0.151	-
	KER	0.318	0.335	0.455	0.304	0.210	-	-
KERP	0.280	0.339	<b>0.482</b>	<b>0.325</b>	<b>0.226</b>	<b>0.162</b>	<b>57.425</b>	
CX-CHR	CNN-RNN	1.580	0.578	0.592	0.506	0.450	0.411	-
	LRCN	1.589	0.577	0.593	0.508	0.459	0.413	-
	AdaAtt	1.568	0.576	0.588	0.505	0.446	0.409	-
	Att2in	1.564	0.576	0.587	0.503	0.447	0.403	25.937
	HRG	2.800	0.588	0.629	0.547	0.497	0.463	-
	HRGR-Agent	<b>2.895</b>	0.612	<b>0.673</b>	0.587	0.530	<b>0.486</b>	-
	KER	0.817	0.552	0.609	0.489	0.400	0.335	-
KERP	2.850	<b>0.618</b>	<b>0.673</b>	<b>0.588</b>	<b>0.532</b>	0.473	<b>67.820</b>	

Table 1: Automatic and human evaluation on IU X-Ray (upper part) and CX-CHR dataset (lower part) compared with CNN-RNN (Vinyals et al. 2015), LRCN (Donahue et al. 2015), AdaAtt (Lu et al. 2017), Att2in (Rennie et al. 2017), CoAtt (Jing, Xie, and Xing 2018), and HRGR-Agent (Li et al. 2018). \* indicates re-training and evaluation on our data split.

frequencies are no less than 3, resulting in 1,479 unique tokens. We collect 155 abnormalities and 362 templates for CX-CHR dataset. More details of the dataset, and collection of abnormalities and templates is in Appendix.

On both dataset, we randomly split the data by patients into training, validation and testing by a ratio of 7:1:2. There is no overlap between patients in different sets. To fairly compare with all baselines, we extract visual features for both dataset from a DenseNet (Huang et al. 2017) jointly pretrained on CX-CHR and public available ChestX-ray8 dataset (Wang et al. 2017). Please see Appendix for details.

**Evaluation metrics.** We use area under the curve (AUC) to evaluate performance of abnormality and disease classification. For evaluating medical report generation, we use two kinds of evaluation metrics: 1) automatic metrics including CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015), ROUGE-L (Lin 2013), BLEU (Papineni et al. 2002); 2) human evaluation: we randomly select 100 sampled testing results of each method, and conduct surveys through Amazon Mechanical Turk. Each survey question gives a ground truth report, and ask candidate to choose among reports generated by different models. The criteria is the degree of correctness of abnormal findings, language fluency, and content coverage compared to the ground truth report. A default choice is provided in case of no or all reports are preferred. We collect results from 5 participants, and compute the average preference percentage for each model excluding default choices.

**Training details.** We first train *Encode* for abnormality classification, then separately train *Retrieve* with fitted templates supervision, and *Paraphrase* with fitted templates as input and ground truth report as output with fixed *Encode*. Then we fine-tune *Paraphrase* using sampled templates from *Retrieve*. As *Retrieve* may not predict same length or same order of sentences as ground truth, we reorder the ground truth sentences by choosing the smallest edit distance to the corresponding retrieved template sen-

Dataset	Model	Abnormality	Disease
IU X-Ray	DenseNet	0.612	0.646
	Ours-1Graph	0.674	-
	Ours-2Graphs	<b>0.686</b>	<b>0.726</b>
CX-CHR	DenseNet	0.689	0.800
	Ours-1Graph	0.721	-
	Ours-2Graphs	<b>0.760</b>	<b>0.862</b>

Table 2: Classification AUC.

tence. We use learning rate  $1e^{-3}$  for training and  $1e^{-5}$  for fine-tuning, and reduce by 10 times when encountering validation performance plateau. We use early stopping, batch size 4 and drop out rate 0.1 for all training.

Besides, as observed from baseline models which overly predict most popular and normal reports for all testing samples, and the fact that most medical reports describe normal cases, we add post-processing to increase the length and comprehensiveness of the generated reports for both datasets while allowing KERP to focus on abnormal findings. Please refer to Appendix for detailed description.

**Baselines for abnormality and disease classification** We compare the performance of abnormality and disease classification with a DenseNet (Huang et al. 2017) trained on classifying the same set of abnormality labels or disease labels. For abnormality classification, we compare our method by solely training on abnormality classification (Ours-1Graph), and jointly training on both abnormality and disease classification (Ours-2Graphs). For disease classification, we directly compare with DenseNet.

**Baselines for medical report generation** On both datasets, we compare with 4 state-of-the-art image captioning models: CNN-RNN (Vinyals et al. 2015), LRCN (Donahue et al. 2015), AdaAtt (Lu et al. 2017), and Att2in (Rennie et al. 2017). We use the same visual features and train/test split on both datasets respectively. On IU X-

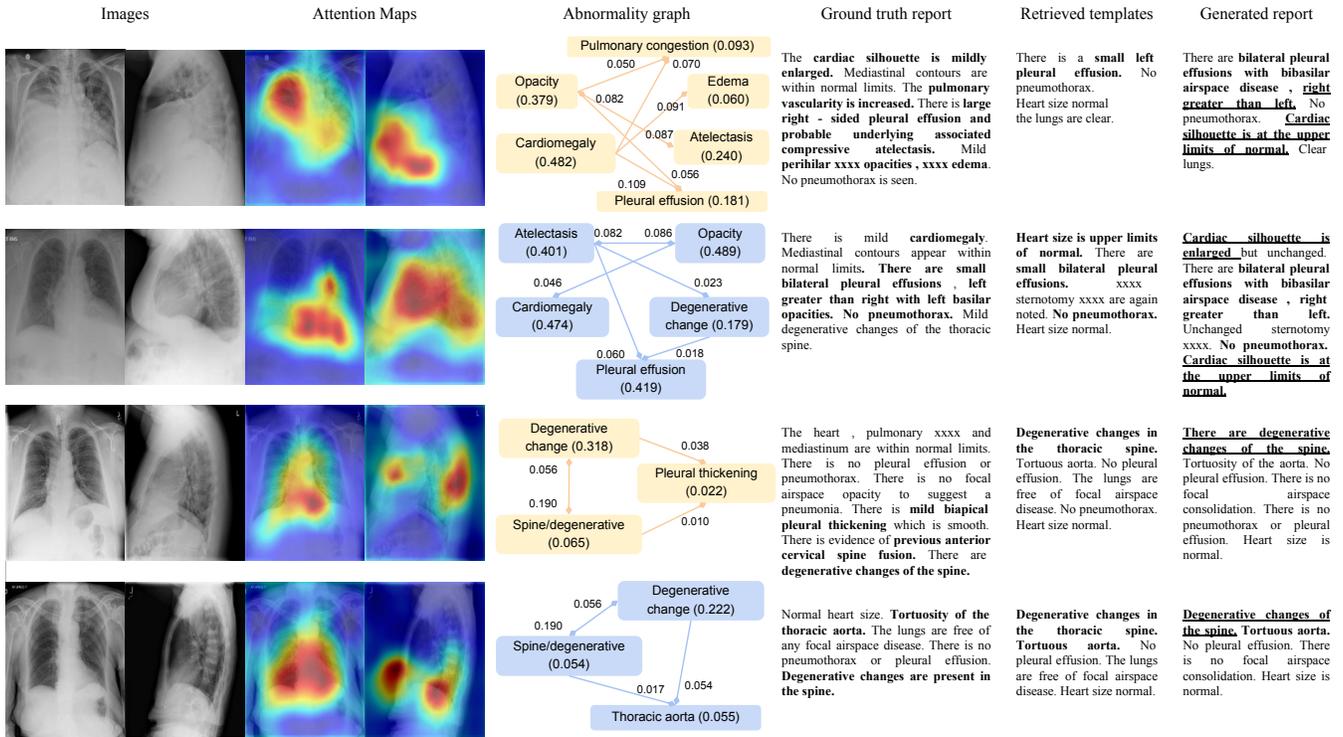


Figure 4: Visualization of result of KERP on IU X-Ray dataset. Bold text indicates alignment between the generated text and ground truth reports. Underlined text indicates correspondence of the generated text (specifically location description) with the visualized attention maps (of either the 1<sub>st</sub>/frontal or 2<sub>nd</sub>/lateral view). In the abnormality graph, values inside parentheses are predicted probabilities of corresponding nodes. We select edges whose attention scores are greater than or equal to 0.05, or are the highest from neighboring nodes to each node, and visualize the attention scores along the directed arrows.

Ray dataset, we also compare with CoAtt (Jing, Xie, and Xing 2018) which uses different visual features extracted from a pretrained ResNet (He et al. 2016). The authors of CoAtt (Jing, Xie, and Xing 2018) re-trained their model using our train/test split, and provided evaluation results for automatic report generation metrics using greedy search and sampling temperature 0.5 at test time. We also compare with HRGR-Agent (Li et al. 2018) which uses different visual features, same set of train/test split for both dataset, and greedy search and argmax sampling at test time. To study the effectiveness of individual modules of KERP, we compare KERP with its variant KER which excludes *Paraphrase* module and only uses retrieved templates as prediction results.

## Results and Analyses

### Abnormality and disease classification error analysis.

The AUCs of abnormality and disease classification are shown in Table 2. Ours-2Graphs outperforms baselines on both dataset on abnormality and disease classification. DenseNet performs the worst on both dataset, and is outperformed by Ours-2Graphs by more than 6%. This demonstrates that, by incorporating prior medical knowledge and learning internal medical knowledge structure, our method is able to distill useful features for correctly classifying ab-

normalities and diseases. Given the high performance of Ours-2Graphs, we conduct following experiments using the knowledge graph trained with both abnormality and disease labels as initialization.

### Medical report generation error analysis.

The error analysis is shown in the upper part of Table 1. Most importantly, KERP outperforms all baselines on BLEU-1,2,3,4 scores on IU X-Ray dataset, and on ROUGE-L, BLEU-1,2,3 scores on CX-CHR dataset, demonstrating its effectiveness and robustness on combining retrieval and generation mechanism for generating medical reports. KERP achieves lower CIDEr score than that of HRGR-Agent on both dataset. However, HRGR-Agent is fine-tuned using reinforcement learning directly with CIDEr as reward. Furthermore, KERP achieves much better performance on abnormality prediction (Table 2 compared to (Li et al. 2018)), demonstrating its superior capability of detecting abnormal findings which is important in clinical diagnosis. Compared to other baseline models that do not use reinforcement learning, KERP obtains the highest CIDEr score on both dataset. On IU X-Ray dataset, KERP achieves lower CIDEr score but higher ROUGE-L and BLEU-n scores than KER which does not have *Paraphrase* process. This indicates that the *Paraphrase* process smooths the retrieved templates into more common sen-

tences as higher BLEU-n scores means higher overlap between n-grams of generated and ground truth reports. However, as CIDEr incorporates inverse document frequency (idf) of words evaluated in the entire evaluation corpus, it inherently gives higher importance to informative and significant phrases, such as abnormal findings and diseases, as oppose to common and popular phrases such as "the lungs are clear" and "heart size is normal". Thus this shows that KER correctly detects more significant phrases, and KERP generates smoother and more common expressions while maintaining the overall performance of abnormal findings detection. On CX-CHR dataset, it is observed that KER performs worse than baseline models in most metrics. This may be due to the fact that the templates for CX-CHR are designed to be concise and to cover large range of different abnormal findings, instead of being natural and common. Thus only using retrieved templates does not lead to high performance. However, the overall high performance of KERP verifies that *Paraphrase* module is able to distill accurate information from the retrieved templates, and paraphrase them into more common and natural descriptions. It also shows that learning conventional and general writing style of radiologists is as important as accurately detecting abnormalities in medical report generation.

**Human evaluation.** We conduct human evaluation as a supplement to automatic evaluation, the result of which is shown in the last column of Table 1. KERP outperforms the compared baseline on both dataset, demonstrating its capability of generating fluent and accurate reports.

**Qualitative analysis.** The visualization result of KERP on IU X-Ray dataset is shown in Figure 4. The generated reports demonstrate significant alignment with ground truth reports as well as correspondence with the visualized attention maps. For example, the generated report of the first sample correctly describes "right greater than left" airspace disease, and the attention map of frontal view shows red region over right upper lung greater than left lower lung (the redness indicates degree of attention). The result demonstrates that KERP is capable of generating accurate reports as well as explainable attentive regions. More visualization and analysis on both dataset is in Appendix.

## Conclusion

We propose a novel Knowledge-driven Encode, Retrieve, Paraphrase (KERP) method to perform accurate and robust medical report generation, and a generic implementation unit-*Graph Transformer* (GTR) which is the first attempt to transform multi-domain graph-structured data via attention mechanism. Experiments show that KERP achieves state-of-the-art performance on two medical image report datasets, and generates accurate attributes prediction, dynamic medical knowledge graph, and explainable location reference.

## References

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-up and top-down attention for image captioning and vqa. In *CVPR*.

Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. 2017. Relational inductive biases, deep learning, and graph networks. *NIPS*.

Bizer, C.; Heath, T.; and Berners-Lee, T. 2011. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *NIPS*.

Demner-Fushman, D.; Kohli, M. D.; Rosenman, M. B.; Shooshan, S. E.; Rodriguez, L.; Antani, S.; Thoma, G. R.; and McDonald, C. J. 2015. Preparing a collection of radiology examinations for distribution and retrieval. *JAMIA*.

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*.

Guu, K.; Hashimoto, T. B.; Oren, Y.; and Liang, P. 2018. Generating sentences by editing prototypes. *AAAI*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Hong, Y., and Kahn, C. E. 2013. Content analysis of reporting templates and free-text radiology reports. *JDI*.

Hu, Z.; Ma, X.; Liu, Z.; Hovy, E.; and Xing, E. 2016. Harnessing deep neural networks with logic rules. In *ACL*.

Hu, Z.; Yang, Z.; Liang, X.; Salakhutdinov, R.; and Xing, E. P. 2017. Toward controlled generation of text. In *ICML*.

Hu, Z.; Yang, Z.; Salakhutdinov, R.; Liang, X.; Qin, L.; Dong, H.; and Xing, E. 2018. Deep generative models with learnable knowledge constraints. *NIPS*.

Huang, T.-H. K.; Ferraro, F.; Mostafazadeh, N.; Misra, I.; Agrawal, A.; Devlin, J.; Girshick, R.; He, X.; Kohli, P.; Batra, D.; et al. 2016. Visual storytelling. In *ACL*.

Huang, G.; Liu, Z.; Weinberger, K. Q.; and van der Maaten, L. 2017. Densely connected convolutional networks. In *CVPR*.

Jing, B.; Xie, P.; and Xing, E. 2018. On the automatic generation of medical imaging reports. *ACL*.

Karpathy, A., and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In *CVPR*.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. *ICLR*.

Krause, J.; Johnson, J.; Krishna, R.; and Fei-Fei, L. 2017. A hierarchical approach for generating descriptive image paragraphs. In *CVPR*.

Li, C. Y.; Liang, X.; Hu, Z.; and Xing, E. P. 2018. Hybrid retrieval-generation reinforced agent for medical image report generation. *NIPS*.

Liang, X.; Hu, Z.; Zhang, H.; Gan, C.; and Xing, E. P. 2017. Recurrent topic-transition gan for visual paragraph generation. In *ICCV*.

Liang, X.; Hu, Z.; and Xing, E. 2018. Symbolic graph reasoning meets convolutions. In *NIPS*.

Lin, T.-Y.; Dollár, P.; Girshick, R. B.; He, K.; Hariharan, B.; and Belongie, S. J. 2017. Feature pyramid networks for object detection. In *CVPR*.

Lin, C.-Y. 2013. Rouge: A package for automatic evaluation of summaries. In *ACL*.

Lu, J.; Xiong, C.; Parikh, D.; and Socher, R. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*.

Lu, J.; Yang, J.; Batra, D.; and Parikh, D. 2018. Neural baby talk. In *CVPR*.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Yang, B.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; et al. 2018. Never-ending learning. *Communications of the ACM*.

Monti, F.; Bronstein, M.; and Bresson, X. 2017. Geometric matrix completion with recurrent multi-graph neural networks. In *NIPS*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, Ł.; Shazeer, N.; and Ku, A. 2018. Image transformer. *ICLR*.

Rennie, S. J.; Marcheret, E.; Mroueh, Y.; Ross, J.; and Goel, V. 2017. Self-critical sequence training for image captioning. In *CVPR*.

Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; and McCallum, A. 2018. Linguistically-informed self-attention for semantic role labeling. In *EMNLP*.

Taskar, B.; Guestrin, C.; and Koller, D. 2004. Max-margin markov networks. In *NIPS*.

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 *NIPS*.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *CVPR*.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2018. Graph attention networks. *ICLR*.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *CVPR*.

Wang, X.; Peng, Y.; Lu, L.; Lu, Z.; Bagheri, M.; and Summers, R. M. 2017. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018a. Non-local neural networks. *CVPR*.

Wang, X.; Peng, Y.; Lu, L.; Lu, Z.; and Summers, R. M. 2018b. Tienet: Text-image embedding network for common thorax disease classification and reporting in chest x-rays. In *CVPR*.

X. Wang, Y. Ye, A. G. 2018. Zero-shot recognition via semantic embeddings and knowledge graphs. *CVPR*.

Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*.

Ziqiang Cao, Wenjie Li, S. L., and Wei, F. 2018. Retrieve, rerank and rewrite: Soft template based neural summarization. In *ACL*.

## Appendices

### Dataset statistics

Detailed statistics of IU X-Ray and CX-CHR dataset is shown in Table 3.

### Visual features extraction

To fairly compare with all baselines, we extract visual features for both dataset from a DenseNet (Huang et al. 2017)

Statistics	IU X-Ray	CX-CHR
#Patients	3,867	35,609
#Images	7,470	45,598
#Diseases	14	14
#Abnormalities	80	155
#Templates	87	362
#Abnormal templates	66	290
#Normal templates	21	72
Vocabulary size	1185	1479
Max. #sentences	18	24
Max. sentence length	42	38
Max. sentence length (by tokens)	42	18
Max. report length	173	216
Avg. #sentences	4.637	9.120
Avg. sentence length	6.997	7.111
Avg. report length	32.450	64.858
Avg. predicted #sentences	4.420	9.984
Avg. predicted sentence length	4.751	7.309
Avg. predicted report length	21.003	66.045

Table 3: Statistics of CX-CHR and IU X-Ray dataset. ”#” indicates ”number of”.

jointly pretrained on CX-CHR and public available ChestX-ray8 dataset (Wang et al. 2017). IU X-Ray dataset is not used for pretraining due to its relatively small size. We add an additional lateral layer as in Feature Pyramid Network (Lin et al. 2017) for the last three dense blocks and additional convolutional layers to transform feature dimension to 256. We extract features from the last convolutional layer of the second dense block which yields  $64 \times 64 \times 256$  feature maps. Such feature maps contain higher resolution and more location information than that directly extracted from the last convolutional layer of a DenseNet (e.g.,  $16 \times 16 \times 1024$ ).

### Abnormality definition & collection

The abnormal findings generally take these forms: 1) the presence of abnormal attributes of an object (e.g., bibasilar consolidation) 2) absence of typical attributes (e.g., disappearance of costophrenic angle) 3) abnormal change of object shape (e.g., enlarged heart size) 4) abnormal change of object location (e.g., elevated left hemidiaphragm). We consider all clinical abnormalities stem from thoracic organs as nodes in the abnormality graph. We use normalized co-occurrence of abnormality attributes computed from training corpus as prior edge weights for the knowledge graph, and normalized co-occurrence of disease labels as prior edge weights for disease graph, normalized co-occurrence of any abnormality attribute and disease labels as edge weights from source knowledge graph nodes to target disease nodes. The coefficients for prior edge weights mentioned above are all set to 0.9.

### Template definition & collection

For each abnormality, we first collect sentences that describe the abnormality and have frequencies no less than a threshold in the training corpus, then manually group sentences of the same meaning and select the most frequent one in each group as template. For generating template sequence and re-

port word sequence, we assume no prior edge weights for the target graph or prior edge weights from source graph to target graph. We assume uniform prior edge weights among all graph nodes, and use 1.0 weight on the learned edge weights.

### **Training details**

$GTR_{i2g}$  for Encode has 3 graph message passing layers and 6 heads in multi-head attention.  $GTR_{g2s}$  and  $GTR_{gs2s}$  for *Retrieve* and *Paraphrase* respectively has 6 graph message passing layers and 8 heads in multi-head attention. The dimension of all hidden features and embedding is set to 256. The coefficients for prior edge weights, if provided, are all set to 0.9. The word embedding of *Retrieve* and *Paraphrase*, and the projection matrix  $\mathbf{W}_w$  to project latent feature to word distribution is shared. We implement our model by PyTorch and train on two GeForce GTX TITAN GPUs.