Writing by Memorizing: Hierarchical Retrieval-based Medical Report Generation

Xingyi Yang
UC San Diego
x3yang@eng.ucsd.edu

Yechao Mu
Penn State University
muchao@psu.edu

Quanzeng You
Microsoft Azure Computer Vision
quyou@microsoft.com

Fenglong Ma
Penn State University
fenglong@psu.edu

Abstract

Medical report generation is one of the most challenging tasks in medical image analysis. Although existing approaches have achieved promising results, they either require a predefined template database in order to retrieve sentences or ignore the hierarchical nature of medical report generation. To address these issues, we propose MedWriter that incorporates a novel hierarchical retrieval mechanism to automatically extract both report and sentence-level templates for clinically accurate report generation. MedWriter first employs the Visual-Language Retrieval (VLR) module to retrieve the most relevant reports for the given images. To guarantee the logical coherence between sentences, the Language-Language Retrieval (LLR) module is introduced to retrieve relevant sentences based on the previous generated description. At last, a language decoder fuses image features and features from retrieved reports and sentences to generate meaningful medical reports. We verified the effectiveness of our model by automatic evaluation and human evaluation on two datasets, i.e., Open-I and MIMIC-CXR

1 Introduction

Medical report generation is the task of generating reports based on medical images, such as radiology and pathology images. Given that this task is time-consuming and cumbersome, researchers endeavor to relieve the burden of physicians by automatically generating the findings and descriptions from medical images with machine learning techniques.

Existing studies can be roughly divided into two categories, i.e., generation-based and retrieval-based approaches. Generation-based methods, including LRCN (Donahue et al., 2015), CoAtt (Jing et al., 2018), and MvH+AttL (Yuan et al., 2019), focus on generating image captions with a encoder-decoder model that leverage image features. However, they are unable to produce linguistically diverse descriptions and depict rare but prominent medical findings. On the other hand, Retrieval-based methods such as HRGR-Agent (Li et al., 2018) and KEPP (Li et al., 2019), pay attention to memorizing templates to generate standardized reports from *a predefined retrieval database*. However, the quality of generated reports significantly depends on the manually curated template database. Besides, they only use sentence-level templates for the generation but ignore to learn the report-level templates, which prevent them from generating more accurate reports.

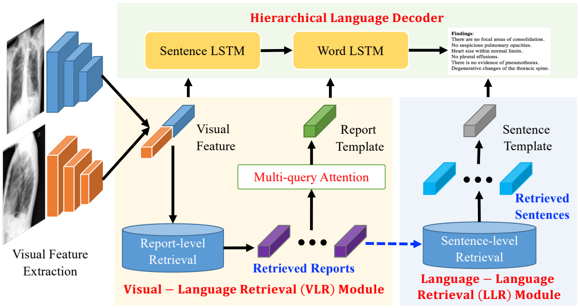

Hierarchical Language Decoder
Sentence LSTM
Word LSTM
Findings:
There are no focal areas of consolidation.
No suspicious pulmonary opacities.
Heart size within normal limits.
No pleural effusions.
There is no evidence of pneumothorax.
Degenerative changes of the thoracic spine.
Visual Feature
Report Template
Sentence Template
Multi-query Attention
Retrieved Sentences
Report-level Retrieval
Retrieved Reports
Sentence-level Retrieval
Visual Feature Extraction
Visual – Language Retrieval (VLR) Module
Language – Language Retrieval (LLR) Module

Figure 1: Overview of the proposed MedWriter.

To address the aforementioned issues, we propose a new framework called MedWriter as shown in Figure 1. MedWriter introduces a novel hierarchical retrieval mechanism working with a hierarchical language decoder to **automatically learn the dynamic report and sentence templates from the data** for generating accurate and professional medical reports. MedWriter is inspired by the process of how physicians write medical reports in real life. They keep report templates in mind and then generate reports for new images by using the key information that they find in the medical images to update the templates sentence by sentence.

In particular, we use three modules to mimic this process. First, MedWriter generates **report-level templates** from the Visual-Language Retrieval (VLR) module using the visual features as the queries. To generate accurate reports, MedWriter also predicts disease labels based on the visual features and extracts medical keywords from the retrieved reports. We propose a **multi-query attention** mechanism to learn the report-level template representations. Second, to make the generated reports more coherent and fluent, we propose a Language-Language Retrieval (LLR) module, which aims to learn **sentence-level templates** for the next sentence generation by analyzing between-sentence correlation in the retrieved reports. Finally, a **hierarchical language decoder** is adopted to generate the full report using visual features, report-level and sentence-level template representations. The designed two-level retrieval mechanism for memorization is helpful in generating accurate and diverse medical reports. To sum up, our contributions are:

- To the best of our knowledge, we are the first to model the memory retrieval mechanism in both report and sentence levels. By imitating the standardized medical report generation in real life, our memory retrieval mechanism effectively utilizes existing templates in the two-layer hierarchy in medical texts. This design allows MedWriter to generate more clinically accurate and standardized reports.
- On top of the retrieval modules, we design a new multi-query attention mechanism to fuse the retrieved information for medical report generation. The fused information can be well incorporated with the existing image and report-level information, which can improve the quality of generated report .
- Experiments conducted on two large-scale medical report generation datasets, i.e., Open-i and MIMIC-CXR show that MedWriter achieves better performance compared with state-of-the-art baselines measured by CIDEr, ROUGE-L, and BLEUs. Besides, case studies show that MedWriter provides more accurate and natural descriptions for medical images through domain expert evaluation.

2 Related work

Generation-based report generation Visual captioning is the process of generating a textual description given an image or a video. The dominant neural network architecture of the captioning task is based on the encoder-decoder framework (Bahdanau et al., 2014; Vinyals et al., 2015; Mao et al., 2014), with attention mechanism (Xu et al., 2015; You et al., 2016; Lu et al., 2017; Anderson et al., 2018; Wang et al., 2019). As a sub-task in the medical domain, early studies directly apply state-of-the-art encoder-decoder models as CNN-RNN (Vinyals et al., 2015), LRCN (Donahue et al., 2015) and AdaAtt (Lu et al., 2017) to medical report generation task. To further improve long text generation with domain-specific knowledge, later generation-based methods introduce hierarchical LSTM with co-attention (Jing et al., 2018) or use the medical concept features (Yuan et al., 2019) to attentively guide the report generation. On the other hand, the concept of reinforcement learning (Liu et al., 2019) is utilized to ensure the generated radiology reports correctly describe the clinical findings.

To avoid generating clinically non-informative reports, external domain knowledge like knowledge graphs (Zhang et al., 2020; Li et al., 2019) and anchor words (Biswal et al., 2020) are utilized to promote the medical values of diagnostic reports. CLARA (Biswal et al., 2020) also provides an interactive solution that integrates the doctors' judgment into the generation process.

Retrieval-based report generation Retrieval-based approaches are usually hybridized with generation-based ones to improve the readability of generated medical reports. For example, KERP (Li et al., 2019) uses abnormality graphs to retrieve most related sentence templates during the generation. HRGR-Agent(Li et al., 2018) incorporates retrieved sentences in a reinforcement learning framework for medical report generation. However, they all require a template database as the model input. Different from these models, MedWriter is able to automatically learn both report-level and sentence-level templates from the data, which significantly enhances the model applicability.

3 Method

As shown in Figure 2, we propose a new framework called MedWriter, which consists of three modules. The **Visual-Language Retrieval (VLR)** module works on the *report level* and uses visual features to find the most relevant template reports based on a multi-view image query. The **Language-Language Retrieval (LLR)** module works on the *sentence level* and retrieves a series of candidates that are most likely to be the next sen-

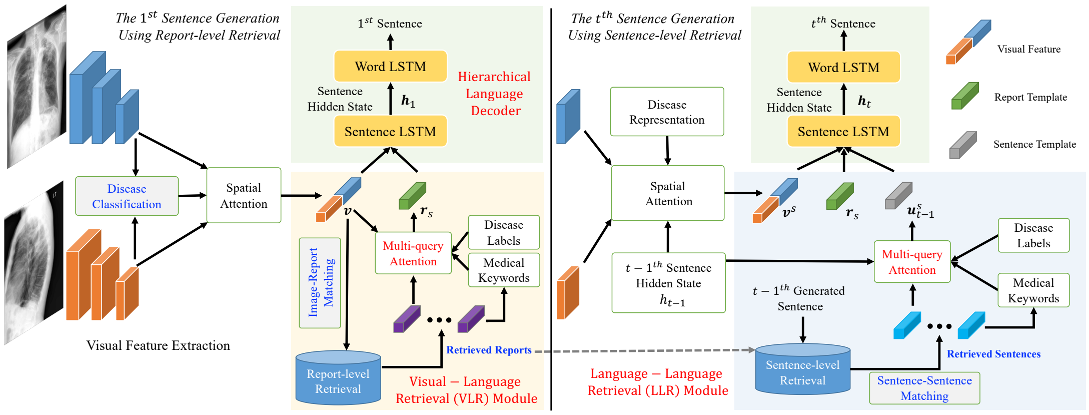


Figure 2: Details of the proposed MedWriter model for medical report generation. The left part is used to learn report template representations via the visual-language retrieval (VLR) module, which is further used to generate the first sentence via the hierarchical language decoder. The right part shows the details of the language-language (LLR) module used for generating the remaining sentences.

tence from the retrieval pool given the generated language context. Finally, MedWriter generates accurate, diverse, and disease-specified medical reports by a **hierarchical language decoder** that fuses the visual, linguistics and pathological information obtained by VLR and LLR modules. To improve the effectiveness and efficiency of retrieval, we first pretrain VLR and LLR modules to build up a retrieval pool for medical report generation as follows.

3.1 VLR module pretraining

The VLR module aims to retrieve the most relevant medical reports from the training report corpus for the given medical images. The retrieved reports are further used to learn an abstract template for generating new high-quality reports. Towards this goal, we introduce a self-supervised pretraining task by judging whether an image-report pair come from the same subject, i.e., **image-report matching**. It is based on an intuitive assumption that an image-report pair from the same subject shares certain common semantics. More importantly, the disease types associated with images and the report should be similar. Thus, in the pretraining task, we also take disease categories into consideration.

3.1.1 Disease classification

The input of the VLR module is a series of multi-modal images and the corresponding report $(\{\boldsymbol{I}_i\}_{i=1}^b, r)$ where the set $\{\boldsymbol{I}_i\}_{i=1}^b$ consists of b images, and r denotes the report. We employ a Convolutional Neural Network (CNN) $f_v(\cdot)$ as the image encoder to obtain the feature of a given image $\boldsymbol{I}_i$, i.e., $\boldsymbol{v}_i = f_v(\boldsymbol{I}_i)$, where $\boldsymbol{v}_i \in \mathbb{R}^{k\times k\times d}$ is the visual feature for the i-th image $\boldsymbol{I}_i$.

With all the extracted features $\{\boldsymbol{v}_i\}_{i=1}^b$, we add them together as the inputs of the disease classification task, which is further used to learn the disease type representation as follows,

$$\boldsymbol{c}_{pred} = \mathbf{W}_{cls}(\sum_{i=1}^{b} \text{AvgPool}(\boldsymbol{v}_i)) + \mathbf{b}_{cls}, \qquad (1)$$

where $\mathbf{W}_{cls} \in \mathbb{R}^{c\times d}$ and $\mathbf{b}_{cls} \in \mathbb{R}^c$ are the weight and bias terms of a linear model, AvgPool is the operation of average pooling, c is the number of disease classes, and $\boldsymbol{c}_{pred} \in \mathbb{R}^c$ can be used to compute disease probabilities as a multi-label classification task with a sigmoid function, i.e., $p_{dc} = \text{sigmoid}(\boldsymbol{c}_{pred})$.

3.1.2 Image-report matching

The next training task for VLR is to predict whether an image-report pair belongs to the same subject. In this subtask, after obtaining the image features $\{\boldsymbol{v}_i\}_{i=1}^b$ and the disease type representation $\boldsymbol{c}_{pred}$, we extract a context visual vector $\boldsymbol{v}$ by the pathological attention.

First, for each image feature $\boldsymbol{v}_i$, we use the disease type representation $\boldsymbol{c}_{pred}$ to learn the spatial attention score through a linear transformation,

$$\boldsymbol{a}_v = \mathbf{W}_a \tanh(\mathbf{W}_v \boldsymbol{v}_i + \mathbf{W}_c \boldsymbol{c}_{pred}) \qquad (2)$$

where $\boldsymbol{a}_v \in \mathbb{R}^{k\times k}$, $\mathbf{W}_a$, $\mathbf{W}_v$ and $\mathbf{W}_c$ are the linear transformation matrices. After that, we use the normalized spatial attention score $\boldsymbol{\alpha}_v = \text{softmax}(\boldsymbol{a}_v)$ to add visual features over all locations (x, y) across the feature map,

$$\boldsymbol{v}'_i = \sum_{\forall x,y} \boldsymbol{\alpha}_v(x, y)\boldsymbol{v}_i(x, y). \qquad (3)$$

Then, we compute the context vector $\boldsymbol{v}$ of the input image set $\{\boldsymbol{I}_i\}_{i=1}^b$ using a linear layer on the concatenation of all the representation $\boldsymbol{v}_i'$, $\boldsymbol{v} = \text{concat}(\boldsymbol{v}_1', \cdots, \boldsymbol{v}_b')\mathbf{W}_f$, where $\mathbf{W}_f \in \mathbb{R}^{bd \times d}$ is the learnable parameter.

For the image-report matching task, we also need a language representation, which is extracted by a BERT (Devlin et al., 2018) model $f_l(\cdot)$ as the language encoder. $f_l(\cdot)$ converts the medical report r into a semantic vector $\boldsymbol{r} = f_l(r) \in \mathbb{R}^d$. Finally, the probability of the input pair $(\{\boldsymbol{I}_i\}_{i=1}^b, r)$ coming from the same subject can be computed as

$$p_{vl} = \text{sigmoid}(\boldsymbol{r}^T \boldsymbol{v}). \tag{4}$$

Given these two sub-tasks, we simultaneously optimize the cross-entropy losses for both disease classification and image-report matching to train the VLR module.

3.2 LLR module pretraining

A medical report usually has some logical characteristics such as describing the patient's medical images in a from-top-to-bottom order. Besides, the preceding and following sentences in a medical report may provide explanations for the same object or concept, or they may have certain juxtaposition, transition and progressive relations. Automatically learning such characteristics should be helpful for `MedWriter` to generate high-quality medical reports. Towards this end, we propose to pretrain a language-language retrieval (LLR) module to search for the most relevant sentences for the next sentence generation. In particular, we introduce a self-supervised pretraining task for LLR to determine if two sentences $\{s_i, s_j\}$ come from the same report, i.e., **sentence-sentence matching**.

Similar to the VLR module, we use a BERT model $f_s(\cdot)$ as the sentence encoder to embed the sentence inputs $\{s_i, s_j\}$ into feature vectors $\boldsymbol{s}_i = f_s(s_i)$, $\boldsymbol{s}_j = f_s(s_j)$. Then the probability that two sentences $\{s_i, s_j\}$ come from the same medical report is measure by

$$p_{ll} = \text{sigmoid}(\boldsymbol{s}_i^{\mathrm{T}} \boldsymbol{s}_j). \tag{5}$$

Again, the cross-entropy loss is used to optimize the learning objective given probability p_{ll} and the ground-truth label of whether s_1 and s_2 belong to the same medical report or not.

3.3 Retrieval-based report generation

Using the pretrained VLR and LLR modules, `MedWriter` generates a medical report given a sequence of input images $\{\boldsymbol{I}_i\}_{i=1}^b$ using a novel hierarchical retrieval mechanism with a hierarchical language decoder.

3.3.1 VLR module for report-level retrieval

Report retrieval Let $\mathcal{D}_r^{(tr)} = \{r_j\}_{j=1}^{N_{tr}}$ denote the set of all the training reports, where N_{tr} is the number of reports in the training dataset. For each report r_j, `MedWriter` first obtain its vector representation using $f_r(\cdot)$ in the VLR module, which is denoted as $\boldsymbol{r}_j = f_r(r_j)$. Let $\mathcal{P}_r = \{\boldsymbol{r}_j\}_{j=1}^{N_{tr}}$ denote the set of training report representations. Given the multi-modal medical images $\{\boldsymbol{I}_i\}_{i=1}^b$ of a subject, the VLR module aims to return the *top k_r medical reports* $\{r_j'\}_{j=1}^{k_r}$ as well as *medical keywords* within in the retrieved reports.

Specifically, `MedWriter` extracts the image feature $\boldsymbol{v}$ for $\{\boldsymbol{I}_i\}_{i=1}^b$ using the pathological attention mechanism as described in Section 3.1. According to Eq. (4), `MedWriter` then computes a image-report matching sore p_{vl} between $\boldsymbol{v}$ and each $\boldsymbol{r} \in \mathcal{P}_r$. The top k_r reports $\{r_j'\}_{j=1}^{k_r}$ with the largest scores p_{vl} are considered as the most relevant medical reports corresponding to the images, and they are selected as the template descriptions. From these templates, we identify n medical keywords $\{w_i\}_{i=1}^n$ using a dictionary as a summarization of the template information. The medical keyword dictionary includes disease phenotype, human organ, and tissue, which consists of 36 medical keywords extracted from the training data with the highest frequency.

Report template representation learning The retrieved reports are highly related to the given images, which should be helpful for the report generation. To make full use of them, we need to learn a report template representation using the image feature $\boldsymbol{v}$, the features of retrieved reports $\{\boldsymbol{r}_j'\}_{j=1}^{k_r}$, medical keywords embeddings $\{\boldsymbol{w}_i\}_{i=1}^n$ for $\{w_i\}_{i=1}^n$ learned from the pretrained word embeddings, and the disease embeddings $\{\boldsymbol{c}_k\}_{k=1}^m$ from predicted disease labels $\{c_k\}_{k=1}^m$ using Disease Classification in Section 3.1.1.

We propose a new **multi-query attention** mechanism to learn the report template representation. To specify, we use the image features $\boldsymbol{v}$ as the key vector $\boldsymbol{K}$, the retrieved report features $\{\boldsymbol{r}_j'\}_{j=1}^{k_r}$ as the value matrix $\boldsymbol{V}$, and the embeddings of both medical keywords $\{\boldsymbol{w}_i\}_{i=1}^n$ and disease labels $\{\boldsymbol{c}_k\}_{k=1}^m$ as the query vectors $\boldsymbol{Q}$. We modify the original self-attention (Vaswani et al., 2017) into a multi-query attention. For each query vector $\boldsymbol{Q}_i$ in $\boldsymbol{Q}$, we first get a corresponding attended feature and then transform them into the **report template vector** $\boldsymbol{r}_s$ after concatenation,

$$\begin{aligned}\boldsymbol{r}_s &= \text{MultiQuery}(\{\boldsymbol{Q}_i\}_{i=1}^{n}, \boldsymbol{K}, \boldsymbol{V}) \\ &= \text{concat}(\text{attn}_1, \cdots, \text{attn}_n)\boldsymbol{W}^O,\end{aligned} \tag{6}$$

where $\text{attn}_i = \text{Attention}(\boldsymbol{Q}_i, \boldsymbol{K}\boldsymbol{W}^K, \boldsymbol{V}\boldsymbol{W}^V)$, and $\boldsymbol{W}^K$, $\boldsymbol{W}^V$ and $\boldsymbol{W}^O$ are the transformation matrices. Generally, the Attention function is calculated by

$$\text{Attention}(\boldsymbol{Q}_g, \boldsymbol{K}_g, \boldsymbol{V}_g) = \text{softmax}(\frac{\boldsymbol{Q}_g \boldsymbol{K}_g{}^{\mathrm{T}}}{\sqrt{d_g}})\boldsymbol{V}_g,$$

where $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$ are queries, keys and values in general case, and d_g is the dimension of the query vector.

3.3.2 LLR module for sentence-level retrieval

Since retrieved reports $\{r_j^t\}_{j=1}^{k_r}$ are highly associated with the input images, the sentence within those reports must contain some instructive pathological information that is helpful for sentence-level generation. Towards this end, we first select sentences from the retrieved reports and then learn sentence-level template representation.

Sentence retrieval We first divide the retrieved reports into L candidate sentences $\{s_j\}_{j=1}^{L}$ as the retrieval pool in the LLR module. Given the pretrained LLR language encoder $f_s(\cdot)$, we can obtain the sentence-level feature pool, which is $\mathcal{P}_s = \{f_s(s_j)\}_{j=1}^{L} = \{\boldsymbol{s}_j\}_{j=1}^{L}$. Assume that the generated sentence at time t is denoted as o_t, and its embedding is $\boldsymbol{o}_t = f_s(o_t)$, which is used to find k_s sentences $\{s_j'\}_{j=1}^{k_s}$ with the highest probabilities p_{ll} from the candidate sentence pool using Eq. (5) in Section 3.2.

Sentence template representation learning Similar to the report template representation, we still use the multi-query attention mechanism. From the retrieved k_s sentences, we extract the medical keywords $\{w_i'\}_{i=1}^{n}$. Besides, we have the predicted disease labels $\{c_k\}_{k=1}^{m}$. Their embeddings are considered as the query vectors. The embeddings of the extracted sentence, i.e., $\{f_s(s_j')\}_{j=1}^{k_s} = \{\boldsymbol{s}_j'\}_{j=1}^{k_s}$, are treated as the value vectors. The key vector is the current sentence (word) hidden state $\boldsymbol{h}_t^s$ ($\boldsymbol{h}_i^w$), which will be introduced in Section 3.3.3. According to Eq. (6), we can obtain the sentence template representation at time t, which is denoted as $\boldsymbol{u}_t$ ($\boldsymbol{u}_i^w$ used for word-level generation).

3.3.3 Hierarchical language decoder

With the extracted features by the retrieval mechanism described above, we apply a hierarchical decoder to generate radiology reports according to the hierarchical linguistics structure of the medical reports. The decoder contains two layers, i.e., a sentence LSTM decoder that outputs sentence hidden states, and a word LSTM decoder which decodes the sentence hidden states into natural languages. In this way, reports are generated sentence by sentence.

Sentence-level LSTM For generating the t-th sentence, `MedWriter` first uses the previous $t-1$ sentences to learn the sentence-level hidden state $\boldsymbol{h}_t^s$. Specifically, `MedWriter` learns the image feature $\boldsymbol{v}^s$ based on Eq. (3). When calculating the attention score with Eq. (2), we consider both the information obtained from the previous $t-1$ sentences (the hidden state $\boldsymbol{h}_{t-1}^s$) and the predicted disease representation from Eq. (1), i.e., replacing $\boldsymbol{c}_{pred}$ with $\text{concat}(\boldsymbol{h}_{t-1}, \boldsymbol{c}_{pred})$. Then the concatenation of the image feature $\boldsymbol{v}^s$, the report template representation $\boldsymbol{r}_s$ from Eq. (6), and the sentence template representation $\boldsymbol{u}_{t-1}^s$ is used as the input of the sentence LSTM to learn the hidden state $\boldsymbol{h}_t^s$

$$\boldsymbol{h}_t^s = \text{LSTM}_s(\text{concat}(\boldsymbol{v}^s, \boldsymbol{u}_{t-1}^s, \boldsymbol{r}_s), \boldsymbol{h}_{t-1}^s), \tag{7}$$

where $\boldsymbol{u}_{t-1}^s$ is obtained using the multi-query attention, the key vector is the hidden state $\boldsymbol{h}_{t-1}^s$, the value vectors are the representations of the retrieved sentences according to the $(t-1)$-th sentence, and the query vectors are the embeddings of both medical keywords extracted from the retrieved sentences and the predicted disease labels.

Word-level LSTM Based on the learned $\boldsymbol{h}_t^s$, `MedWriter` conducts the word-by-word generation using a word-level LSTM. For generating the $(i+1)$-th word, `MedWriter` first learns the image feature $\boldsymbol{v}^w$ using Eq. (2) by replacing $\boldsymbol{c}_{pred}$ with $\boldsymbol{h}_i^w$ in Eq. (2), where $\boldsymbol{h}_i^w$ is the hidden state of the i-th word. `MedWriter` then learns the sentence template representation $\boldsymbol{u}_i^w$ using the multi-query attention, where the key vector is the hidden state $\boldsymbol{h}_i^w$, value and query vectors are the same as those used for calculating $\boldsymbol{u}_{t-1}^s$. Finally, the concatenation of $\boldsymbol{h}_t^s$, $\boldsymbol{u}_i^w$, $\boldsymbol{v}^w$, and $\boldsymbol{r}_s$ is taken as the input of the word-level LSTM to generate the $(i+1)$-th word as follows:

$$\begin{aligned}&\boldsymbol{h}_i^w = \text{LSTM}_w(\text{concat}(\boldsymbol{h}_t^s, \boldsymbol{u}_i^w, \boldsymbol{v}^w, \boldsymbol{r}_s), \boldsymbol{h}_{i-1}^w), \\ &w_{i+1} = \text{argmax}(\text{softmax}(FFN(\boldsymbol{h}_i^w))),\end{aligned} \tag{8}$$

where $FFN(\cdot)$ is the feed-forward network.

Note that for the first sentence generation, we set $\boldsymbol{u}_0$ as $\mathbf{0}$, and $\boldsymbol{h}_0$ is the randomly initialized vector, to learn the sentence-level hidden state $\boldsymbol{h}_1^s$. When generating the words of the first sentence, we set $\boldsymbol{u}_i^w$ as the $\mathbf{0}$ vector.

4 Experiments

4.1 Datasets and baselines

Datasets **Open-i**[1] (Demner-Fushman et al., 2016) (*a.k.a* IU X-Ray) provides 7,470 chest X-rays with 3,955 radiology reports. In our experiments, we only utilize samples with both frontal and lateral views, and with complete findings and impression sections in the reports. This results in totally 2,902 cases and 5,804 images. **MIMIC-CXR**[2] (Johnson et al., 2019) contains 377,110 chest X-rays associated with 227,827 radiology reports, divided into subsets. We use the same criterion to select samples, which results in 71,386 reports and 142,772 images.

For both datasets, we tokenize all words with more than 3 occurrences and obtain 1,252 tokens on the Open-i dataset and 4,073 tokens on the MIMIC-CXR dataset, including four special tokens ⟨PAD⟩, ⟨START⟩, ⟨END⟩, and ⟨UNK⟩. The findings and impression sections are concatenated as the ground-truth reports. We randomly divide the whole datasets into train/validation/test sets with a ratio of 0.7/0.1/0.2. To conduct the disease classification task, we include 20 most frequent finding keywords extracted from MeSH tags as disease categories on the Open-i dataset and 14 CheXpert categories on the MIMIC-CXR dataset.

Baselines On both datasets, we compare with four state-of-the-art image captioning models: CNN-RNN (Vinyals et al., 2015), CoAttn (Jing et al., 2018), MvH+AttL (Yuan et al., 2019), and V-L Retrieval. V-L Retrieval only uses the retrieved report templates with the highest probability as prediction without the generation part based on our pretrained VLR module. Due to the lack of the opensource code for (Wang et al., 2018; Li et al., 2019, 2018; Donahue et al., 2015) and the template databases for (Li et al., 2019, 2018), we only include the *reported results* on the Open-i dataset in our experiments.

4.2 Experimental setup

All input images are resized to 512×512, and the feature map from DenseNet-121 (Huang et al., 2017) is $1024 \times 16 \times 16$. During training, we use random cropping and color histogram equalization for data augmentation.

To pretrain the VLR module, the maximum length of the report is restricted to 128 words. We train VLR module for 100 epochs with an Adam (Kingma and Ba, 2014) optimizer with 1e-5 as the initial learning rate, 1e-5 for L2 regularization, and 16 as the mini-batch size. To pretrain the LLR module, the maximum length of each sentence is set to 32 words. We optimize the LLR module for 100 epochs with an Adam (Kingma and Ba, 2014) optimizer with the initial learning rate of 1e-5 and a mini-batch size of 64. The learning rate is multiplied by 0.2 every 20 epochs.

To train the full model for `MedWriter`, we set the retrieved reports number $k_r = 5$ and sentences number $k_s = 5$. Extracting $n = 5$ medical keywords and predicting $m = 5$ disease labels are used for report generation. Both sentence and word LSTM have 512 hidden units. We freeze the weights for the pretrained VLR and LLR modules and only optimize on the language decoder. We set the initial learning rate as 3e-4 and mini-batch size as 32. `MedWriter` takes 10 hours to train on the Open-i dataset and 3 days on the MIMIC-CXR dataset with four GeForce GTX 1080 Ti GPUs.

4.3 Quantitative and qualitative results

Table 1 shows the CIDEr, ROUGE-L, BLUE, and AUC scores achieved by different methods on the test sets of Open-i and MIMIC-CXR.

Language evaluation From Table 1, we make the following observations. First, compared with *Generation*-based model, *Retrieval*-based model that uses the template reports as results has set up a relatively strong baseline for medical report generation. Second, compared with V-L retrieval, other *Retrieval*-based approaches perform much better in terms of all the metrics. This again shows that that by integrating the information retrieval method into the deep sequence generation framework, we can not only use the retrieved language information as templates to help generate long sentences, but also overcome the monotony of only using the templates as the generations. Finally, we see that the proposed `MedWriter` achieves the highest language scores on 5/6 metrics on Open-i

[1] https://openi.nlm.nih.gov/faq#collection

[2] https://physionet.org/content/mimic-cxr/2.0.0/

Dataset	Type	Model	CIDEr	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4	AUC
Open-i	Generation	CNN-RNN (Vinyals et al., 2015)	0.294	0.307	0.216	0.124	0.087	0.066	0.426
		LRCN (Donahue et al., 2015)*	0.285	0.307	0.223	0.128	0.089	0.068	–
		Tie-Net (Wang et al., 2018)*	0.279	0.226	0.286	0.160	0.104	0.074	–
		CoAtt (Jing et al., 2018)	0.277	0.369	0.455	0.288	0.205	0.154	0.707
		MvH+AttL (Yuan et al., 2019)	0.229	0.351	0.452	0.311	0.223	0.162	0.725
	Retrieval	V-L Retrieval	0.144	0.319	0.390	0.237	0.154	0.105	0.634
		HRGR-Agent (Li et al., 2018)*	0.343	0.322	0.438	0.298	0.208	0.151	–
		KERP (Li et al., 2019)*	0.280	0.339	**0.482**	0.325	0.226	0.162	–
		`MedWriter`	**0.345**	**0.382**	0.471	**0.336**	**0.238**	**0.166**	**0.814**
		Ground Truth	–	–	–	–	–	–	0.915
MIMIC-CXR	Generation	CNN-RNN (Vinyals et al., 2015)	0.245	0.314	0.247	0.165	0.124	0.098	0.472
		CoAtt (Jing et al., 2018)	0.234	0.274	0.410	0.267	0.189	0.144	0.745
		MvH+AttL (Yuan et al., 2019)	0.264	0.309	0.424	0.282	0.203	0.153	0.738
	Retrieval	V-L Retrieval	0.186	0.232	0.306	0.179	0.116	0.076	0.579
		`MedWriter`	**0.306**	**0.332**	**0.438**	**0.297**	**0.216**	**0.164**	**0.833**
		Ground Truth	–	–	–	–	–	–	0.923

Table 1: Automatic evaluation on the Open-i and MIMIC-CXR datasets. * indicates the results reported in (Li et al., 2019).

datasets and all metrics on MIMIC-CXR among all methods. `MedWriter` not only improves current SOTA model CoAttn (Jing et al., 2018) by 5% and MvH+AttL (Yuan et al., 2019) by 4% on Open-i in average, but also goes beyond SOAT retrieval-based approaches like KERP (Li et al., 2019) and HRGR-Agent (Li et al., 2018) and significantly improves the performance, even *without using manually curated template databases*. This illustrates the effectiveness of automatically learning templates and adopting hierarchical retrieval in writing medical reports.

Clinical evaluation We train two report classification BERT models on both datasets and use it to judge whether the generated reports correctly reflect the ground-truth findings. We show the mean AUC scores achieved by ground-truth and generated reports from different baselines in the last column of Table 1. We can observe that `MedWriter` achieves the highest AUC scores compared with other baselines. In addition, our method achieves the AUC scores that are very close to those of professional doctors' reports, with 0.814/0.915 and 0.833/0.923 on two datasets. This shows that the generation performance of `MedWriter` has approached the level of human domain experts, and it embraces great medical potentials in identifying disease-related medical findings.

Human evaluation We also qualitatively evaluate the quality of the generated reports via a user study. We randomly select 50 samples from the Open-i test set and collect ground-truth reports and the generated reports from both MvH+AttL (Yuan et al., 2019) and `MedWriter` to conduct the human evaluation. Two experienced radiologists were asked to give ratings for each selected report, in terms of whether the generated reports are realistic and relevant to the X-ray images. The ratings are integers from one to five. The higher, the better.

Table 2 shows average human evaluation results on `MedWriter` compared with Ground Truth reports and generations of MvH+AttL (Yuan et al., 2019) on Open-i, evaluated in terms of realistic scores and relevant scores. `MedWriter` achieves much higher human preference than the baseline model, even approaching the performance of Ground Truth reports that wrote by experienced radiologists. It shows that `MedWriter` is able to generate accurate clinical reports that are comparable to domain experts.

Method	Realistic Score	Relevant Score
Ground Truth	3.85	3.82
MvH+AttL (Yuan et al., 2019)	2.50	2.57
`MedWriter`	3.68	3.44

Table 2: User study conducted by two domain experts.

Qualitative analysis Figure 3 shows qualitative results of `MedWriter` and baseline models on the Open-i dataset. `MedWriter` not only produces longer reports compared with MvH+AttL but also accurately detects the medical findings in the images (marked in red and **bold**). On the other hand, we find that `MedWriter` is able to put forward some supplementary suggestions (marked in blue) and descriptions, which are not in the original report but have diagnostic value. The underlying reason for this merit comes from the memory retrieval mechanism that introduces prior medical knowledge to facilitate the generation process.

4.4 Ablation study

We perform ablation studies on the Open-i and MIMIC-CXR datasets to investigate the effectiveness of each module in `MedWriter`. In each of the following studies, we change one module with other modules intact.

Removing the VLR module In this experiment, global report feature $\boldsymbol{r}_s$ is neglected in Eqs. (7)

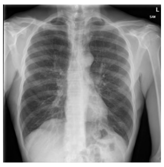
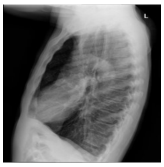
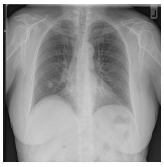
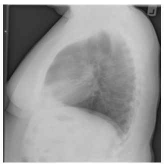
Frontal Image	Lateral Image	Ground Truth	MvH+AttL	MedWriter
		emphysematous changes. resolution of prior right midlung infiltrate. previous ⟨UNK⟩ is normal in size and contour. lungs are clear. **no focal consolidation pneumothorax or pleural effusion.** interval ⟨UNK⟩ of previously described right midlung opacity suggesting resolved ⟨UNK⟩ process. lungs are hyperexpanded with flattened diaphragms. ⟨UNK⟩ and soft tissue are unremarkable.	no acute cardiopulmonary disease. the heart is normal in size. the lungs are clear. **there is no pleural effusion or pneumothorax.** of the right clavicle. ⟨UNK⟩ ⟨UNK⟩ are present. ⟨UNK⟩ to the glenoid joints.	hyperexpanded lungs. ⟨UNK⟩ right upper lobe ⟨UNK⟩. no focal pneumonia. the cardiomediastinal silhouette is normal in size and contour. **negative for focal consolidation pneumothorax or large pleural effusion.** negative for acute bone abnormality.
		chest. large nodule at the right lung base that probably represents a granuloma although not it is not ⟨UNK⟩ calcified. there is a ⟨UNK⟩ mm nodule in the right lower lobe that is relatively dense but not ⟨UNK⟩ calcified on the corresponding rib series. there are probably right hilar calcified lymph ⟨UNK⟩. lungs otherwise are clear. **there is no pleural effusion**. left ribs. *no fracture or focal bony destruction.*	no acute cardiopulmonary disease. the heart is normal in size and contour. are clear without evidence of infiltrate. **is no pneumothorax**. *degenerative changes of the thoracic spine..* head..	right upper lobe pneumonia. consideration may be given for primary or ⟨UNK⟩. recommend ct of the chest may be helpful for further diagnosis. in the interval a 3 cm ⟨UNK⟩ mass has developed in the right lower lobe. **no pneumothorax or pleural effusion**. the mediastinal contours are normal.

Figure 3: Examples of ground-truth and generated reports by MvH+AttL (Yuan et al., 2019) and MedWriter. Highlighted red phrases are medical abnormality terms that generated and ground-truth reports have in common. **Bold** terms are common descriptions of normal tissues. The text in *italics* is the opposite meaning of the generated report and the actual report. We mark the supplementary comments to the original report in blue.

Dataset	Model	CIDEr	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Open-i	MedWriter w/o VLRM	0.333	0.373	0.466	0.324	0.229	0.159
	MedWriter w/o LLRM	0.329	0.354	0.453	0.307	0.215	0.154
	MedWriter w/o HLD	0.284	0.317	0.434	0.295	0.208	0.149
	MedWriter	**0.345**	**0.382**	**0.471**	**0.336**	**0.238**	**0.166**
MIMIC-CXR	MedWriter w/o VLRM	0.294	0.317	0.432	0.288	0.209	0.161
	MedWriter w/o LLRM	0.283	0.305	0.425	0.280	0.204	0.157
	MedWriter w/o HLD	0.263	0.287	0.418	0.265	0.187	0.146
	MedWriter	**0.306**	0.**332**	**0.438**	**0.297**	**0.216**	**0.164**

Table 3: Ablation study on both Open-i and MIMIC-CXR datasets.

and (8), and the first sentence is generated only based on image features. The LLR module keeps its functionality. However, instead of looking for sentence-level templates from the retrieved reports, it searches for most relevant sentences from *all the reports*. As can be seen from Table 3, removing VLR module ("w/o VLRM") leads to performance reduction by 2% on average. This demonstrates that visual-language retrieval is capable in sketching out the linguistic structure of the whole report. The rest of the language generation is largely influenced by report-level context information.

Removing the LLR module The generation of $(t+1)$-th sentence is based on the global report feature $\boldsymbol{r}_s$ and the image feature $\boldsymbol{v}$, without using the retrieved sentences information in Eq. (8). Table 3 shows that removing LLR module ("w/o LLRM") results in the decease of average evaluation scores by 4% compared with the full model. This verifies that the LLR module plays an essential role in generating long and coherent clinical reports.

Replacing hierarchical language decoder We use a single layer LSTM that treats the whole report as a long sentence and conduct the generation word-by-word. Table 3 shows that replacing hierarchical language decoder with a single-layer LSTM ("w/o HLD") introduces dramatic performance reduction. This phenomenon shows that the hierarchical generative model can effectively and greatly improve the performance of long text generation tasks.

5 Conclusions

Automatically generating accurate reports from medical images is a key challenge in medical image analysis. In this paper, we propose a novel model named MedWriter to solve this problem based on hierarchical retrieval techniques. In particular, MedWriter consists of three main modules, which are the visual-language retrieval (VLR) module, the language-language retrieval (LLR) module, and the hierarchical language decoder. These three modules tightly work with each other to automatically generate medical reports. Experimental results on two datasets demonstrate the effectiveness of the proposed MedWriter. Besides, qualitative studies show that MedWriter is able to generate meaningful and realistic medical reports.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Siddharth Biswal, Cao Xiao, Lucas Glass, Brandon Westover, and Jimeng Sun. 2020. Clara: Clinical report auto-completion. In *Proceedings of The Web Conference 2020*, pages 541–550.

Dina Demner-Fushman, Marc D Kohli, Marc B Rosenman, Sonya E Shooshan, Laritza Rodriguez, Sameer Antani, George R Thoma, and Clement J McDonald. 2016. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2):304–310.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Baoyu Jing, Pengtao Xie, and Eric Xing. 2018. On the automatic generation of medical imaging reports. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2577–2586.

Alistair EW Johnson, Tom J Pollard, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Yifan Peng, Zhiyong Lu, Roger G Mark, Seth J Berkowitz, and Steven Horng. 2019. Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs. *arXiv preprint arXiv:1901.07042*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Christy Y Li, Xiaodan Liang, Zhiting Hu, and Eric P Xing. 2019. Knowledge-driven encode, retrieve, paraphrase for medical image report generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6666–6673.

Yuan Li, Xiaodan Liang, Zhiting Hu, and Eric P Xing. 2018. Hybrid retrieval-generation reinforced agent for medical image report generation. In *Advances in neural information processing systems*, pages 1530–1540.

Guanxiong Liu, Tzu-Ming Harry Hsu, Matthew McDermott, Willie Boag, Wei-Hung Weng, Peter Szolovits, and Marzyeh Ghassemi. 2019. Clinically accurate chest x-ray report generation. *arXiv preprint arXiv:1904.02633*.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383.

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. 2014. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Weixuan Wang, Zhihong Chen, and Haifeng Hu. 2019. Hierarchical attention network for image captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8957–8964.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, and Ronald M Summers. 2018. Tienet: Text-image embedding network for common thorax disease classification and reporting in chest x-rays. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9049–9058.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659.

Jianbo Yuan, Haofu Liao, Rui Luo, and Jiebo Luo. 2019. Automatic radiology report generation based on multi-view image fusion and medical concept enrichment. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 721–729. Springer.

Yixiao Zhang, Xiaosong Wang, Ziyue Xu, Qihang Yu, Alan L. Yuille, and Daguang Xu. 2020. When radiology report generation meets knowledge graph. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 12910–12917.