Cross-Modal Memory Networks for Radiology Report Generation and Disease Classification

Ali Raza[1]
Muhammad Sameed[2]
Syed Saadullah Hussaini[3]

BS(AI) Fall 2024

National University of Computer and Emerging Sciences (FAST)

December 10, 2024

1. Introduction

Radiology report generation plays a critical role in clinical diagnostics by summarizing findings from medical images, such as X-rays, in a structured format. While automated systems have made progress, significant challenges remain:

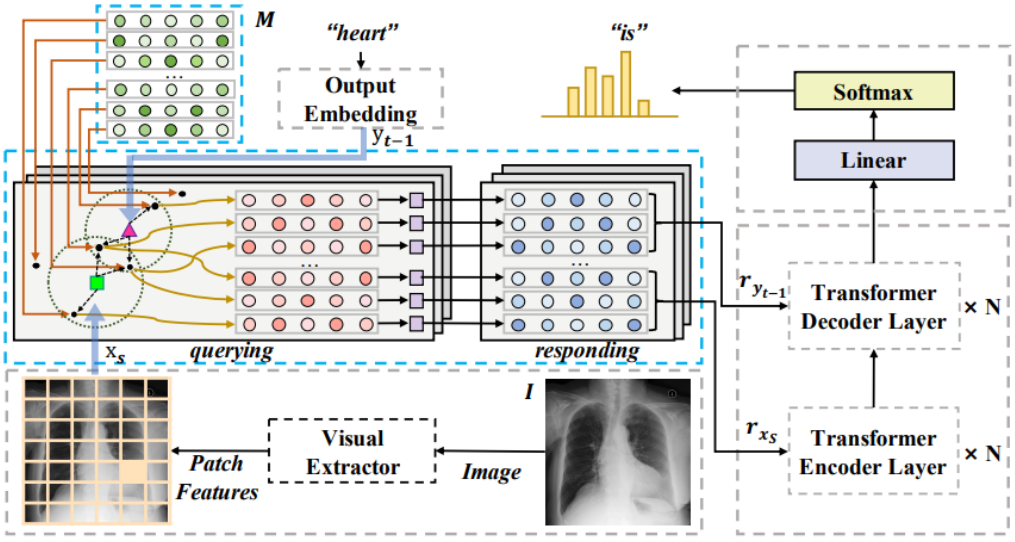
- **Extracting Important Features:** Identifying subtle and critical parts of medical images, such as a lung nodule or fractured bone, is often difficult.
- **Feature Degradation:** Ensuring the preservation of extracted features during the processing stages is crucial to avoid information loss.
- **Focusing Attention:** Current systems often fail to pay adequate attention to the most relevant regions in an image.
- **Cross-Modal Alignment:** Properly aligning visual features with corresponding textual descriptions is another challenge, especially with limited annotated data.

Our research builds on the architecture of Cross-Modal Memory Networks (CMN), addressing the limitations of prior methods to enhance radiology report generation.

2. Base Architecture: Cross-Modal Memory Networks

The CMN architecture focuses on improving cross-modal alignment by introducing a shared memory mechanism to enhance interaction between image and text features. Key elements include:

- **Shared Memory:** Stores alignment information between images and textual descriptions
- **Memory Querying and Responding:** Extracts the most relevant memory vectors and uses them to improve both the encoder and decoder stages of the radiology report generation process.
- **Encoder-Decoder Framework:** Combines convolutional encoders for image feature extraction and sequence-based decoders for text generation.



3. Architectural Enhancements

We improved the CMN architecture by integrating Linear Projection, Attention Pooling, L2 Normalization, and Contrastive Loss to enhance cross-modal alignment.

3.1. Linear Projection Layer

Objective: The primary goal of the Linear Projection Layer is to ensure that the dimensionality of the textual features matches the dimensionality of the visual features before any further processing. This alignment is critical for effective cross-modal fusion, as both feature sets need to be of the same size for comparison and subsequent operations.

Implementation: Given the textual features TTT with a shape of the projection layer maps them to a new feature space that matches the visual features' size . This is done via a linear transformation, formulated as:

$$T' = T \cdot W_t + b_t$$

Where:

- T: Textual feature tensor with shape $(\text{batch\_size}, \text{max\_len}, \text{vocab+1})$.
- W_t: Weight matrix of shape $(\text{vocab+1}, \text{n\_features})$.
- b_t: Bias term.
- T': Projected tensor of shape $(\text{batch\_size}, \text{max\_len}, \text{n\_features})$, matching visual feature dimensions.

This projection ensures that the textual features can now be directly compared with the visual features, which are also in the (batch-size, n-features) shape.

3.2. Attention Pooling Layer

Objective: To focus on the most relevant parts of the textual features, we applied an Attention Pooling Layer. This mechanism allows the model to assign more importance to key textual features while generating a context-aware representation.

Implementation: The attention pooling is applied to the projected textual features Twith shape (batch-size,max-len,n-features)The goal is to reduce the sequence length max-len while preserving the important features, producing a fixed-size vector representation for the text. We compute the attention scores for each token iii in the sequence using a dot product with a learnable attention vector wt:

$$e_i = w_t^\top \cdot T_i'$$

Where:

- T_i': i-th token's feature vector.
- w_t: Learnable attention weight vector.

Next, we apply a softmax function to these scores to get the attention weights.

3.3. L2 Normalization

Objective: To make both the textual and visual features comparable in scale, we apply L2 normalization. This step ensures that the cosine similarity between the features is computed effectively, as all vectors will have a unit length.

Formula:

$$x_{\text{norm}} = \frac{x}{\|x\|}$$

Where $\|x\|$ is the L2 norm of feature vector x and xnorm is the normalized vector. Both the textual features and visual features V are normalized to ensure they are on the same scale, allowing for accurate similarity computations.

3.4. Contrastive Loss

Objective: To encourage better alignment between the visual and textual features, we applied Contrastive Loss. This loss function drives the model to minimize the distance between matching image-text pairs (positive pairs) and maximize the distance between non-matching pairs (negative pairs).

Formula:

$$L_{\text{contrastive}}(v, t) = 1 - \text{Cosine Similarity}(v, t) = 1 - \frac{v \cdot t}{\|v\|\|t\|}$$

Where v and t are normalized visual and textual feature vectors. For negative pairs, the model tries to minimize the cosine similarity, pushing dissimilar pairs farther apart in the feature space.

3.5. Definition of Similar and Dissimilar Pairs

- **Similar Pairs:** A pair of visual and textual features is considered similar if both the visual feature (from an X-ray image) and the textual feature (from the corresponding radiology report) refer to the same report ID. This means the image and text describe the same medical findings. The objective is to minimize the distance between the visual and textual features for these similar pairs, ensuring the image and report are closely aligned in the feature space.
- **Dissimilar Pairs:** A pair is considered dissimilar if the visual feature from one report is paired with a textual feature from a different report. For example, an X-ray image from one patient paired with a radiology report from another patient. The aim for dissimilar pairs is to maximize the distance between the visual and textual features, as these pairs are irrelevant to one another.

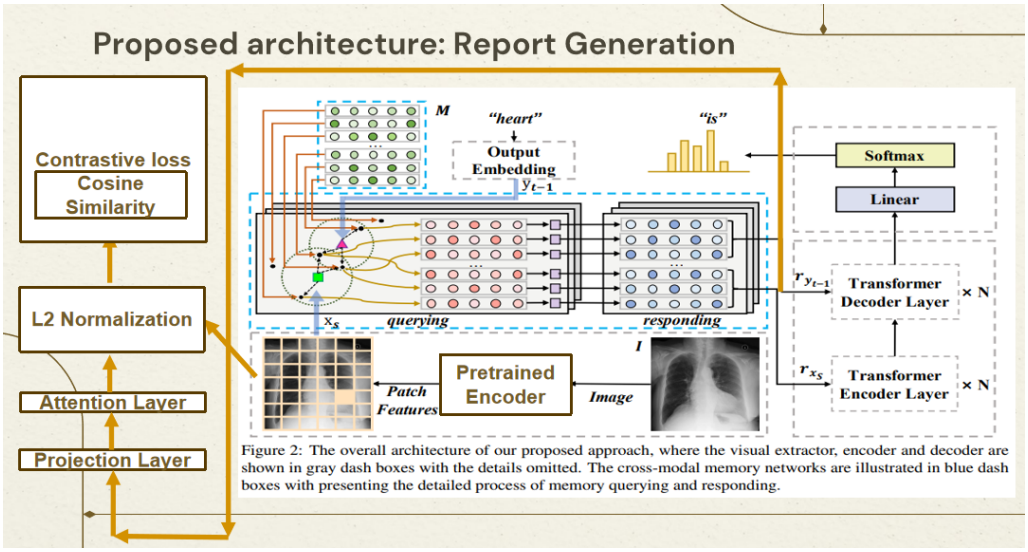


Figure 2: The overall architecture of our proposed approach, where the visual extractor, encoder and decoder are shown in gray dash boxes with the details omitted. The cross-modal memory networks are illustrated in blue dash boxes with presenting the detailed process of memory querying and responding.

4. Disease Classification

For disease classification, we used a pretrained encoder to extract features from both X-ray images (visual) and radiology reports (textual). These features were then fused into a combined representation and passed through a Multi-Layer Perceptron (MLP) to predict disease classes. To address class imbalance, we incorporated focal loss into the standard cross-entropy loss. Focal loss down-weights well-classified examples and focuses on hard-to-classify ones, improving the model's performance on rare diseases. The final loss is a weighted sum of cross-entropy and focal loss, ensuring the model efficiently learns to classify diseases from both modalities.

Combined Loss Function:

$$L_{\text{total}} = L_{\text{CE}} + L_{\text{focal}}$$

Where:

$$L_{\text{CE}} = -\sum_{i} y_i \log(p_i)$$

- y_i: True label for class i.
- p_i: Predicted probability for class i.

5. Results

5.1. EKAGen Results

The EKAGen results on the IU-Xray dataset using different backbones are shown below:

Table 1. EKAGen Results on IU-Xray Dataset

Model	BL-1	BL-2	BL-3	BL-4	MTR	RG-L
EKAGen (ViT-B/16)	0.517	0.351	0.258	0.191	0.211	0.409
EKAGen (RN-101)	0.526	0.361	0.267	0.203	0.214	0.404

5.2. R2Gen CMN Results

The R2Gen CMN results on the IU-Xray dataset for different configurations are shown below:

Table 2. R2Gen CMN Results on IU-Xray Dataset

Data	Model	BL-1	BL-2	BL-3	BL-4	MTR	RG-L
IU-Xray	Base	0.396	0.254	0.179	0.135	0.164	0.342
IU-Xray	+MEM	0.443	0.270	0.191	0.144	0.172	0.351
IU-Xray	+CMN	**0.475**	**0.309**	**0.222**	**0.170**	**0.191**	**0.375**

5.3. Our Proposed Architecture Results on Radiology Report Generation

Table 3. Final Report Generation Metrics

Metric	Score
BLEU-1	0.4788
BLEU-2	0.3024
BLEU-3	0.2167
BLEU-4	0.1607
METEOR	0.1941
ROUGE-L	0.3667

5.4. Our Proposed Architecture Results on Disease Classification

Table 4. Classification Results (With and Without Focal Loss)

Metric	With Focal Loss	Without Focal Loss
Precision (weighted)	80.39%	77.45%
Recall (weighted)	69.09%	69.18%
F1 Score (weighted)	62.16%	62.87%

6. Conclusion

Our modifications to the CMN architecture address key challenges in radiology report generation, including cross-modal alignment, feature degradation, and class imbalance. By introducing a pretrained encoder, focal loss, and refined attention mechanisms, our model:

- Outperforms the baseline CMN and EKAGen architectures in report generation tasks.
- Achieves robust classification metrics, particularly for rare diseases.
- Demonstrates the importance of aligning image and textual features using shared memory for clinical task

These improvements highlight the potential for further optimization and deployment in clinical workflows to reduce radiologists' workload while maintaining diagnostic accuracy.