Medical Report Generation using Chest X Ray Images

Individual Project Report (Team 6)

for the course

DATS 6312 ‘Natural Language Processing’

Fall 2024

Abishek Chiffon Muthu Raja

G31327889

INTRODUCTION

In modern healthcare, radiology plays a crucial role in diagnosing and managing a wide array of medical conditions. Chest X-rays, in particular, are among the most frequently used diagnostic tools for detecting abnormalities such as Pneumonia, Hernia and Cardiomegaly. The motivation behind this project is to create a tool that supports radiologists in their demanding roles by automating the generation of preliminary radiology reports. By leveraging advanced computer vision and large language models, the system aims to draft accurate and detailed reports from chest X-ray images, serving as a starting point for radiologists to review and refine. This

collaboration between AI and radiologists enhances productivity, reduces delays, and minimizes the risk of errors stemming from workload fatigue.

The individual report begins with an **Introduction** that outlines the motivation and scope of the project, followed by a detailed description of the dataset, highlighting its structure and features. The **Methodology** section explains the technical approach, including preprocessing steps, models used, and training processes. My **Contributions section** explains my contributions to the project. The **Results** section presents the performance outcomes, supported by relevant metrics and analysis. The **Conclusions** summarize the key findings from the project. The **Further Improvements** section explores potential advancements, such as integrating additional datasets to improve accuracy and usability. Finally, the **References** provide a comprehensive list of all sources cited throughout the report.

DESCRIPTION OF THE DATASET

The MIMIC-CXR is a publicly available chest X-ray dataset for chest radiography research.It comprises 15,000 chest X-ray images in dicom format and their associated radiology reports in xml format. The dataset has the following key features:

- **Image File Path:** Location or link to the corresponding chest X-ray image.
- **Findings:** A textual description of abnormalities or observations made by the radiologist.
- **Impression:** A concise summary of the radiologist's primary conclusions.

The dataset has 14 labels corresponding to common chest X-ray pathologies.The pathology labels include Atelectasis, Cardiomegaly, Consolidation, Edema, Enlarged Cardiomediastinum, Fracture, Lung Lesion, Lung Opacity, Pleural Effusion, Pleural Other, Pneumonia, Pneumothorax, Support Devices, and No Finding.

METHODOLOGY

The project was structured in several key stages, each critical to the successful implementation of the model:

1. **Data Collection and Preprocessing:** The initial step involved converting DICOM images to PNG format, extracting relevant information, preparing datasets for further analysis, and performing text and image transformations.
2. **Label Extraction:** The next stage utilized ChexBERT to generate multi-label classifications from the associated radiology reports, enabling the identification of relevant medical conditions in each image.
3. **Model Architecture Design:** During this stage, different combinations of image encoders, alignment models, and language models were experimented with to determine the most effective architecture for processing medical image data and generating accurate reports.
4. **Training and Evaluation:** Once the model architecture was defined, the models were trained using the preprocessed data. Performance was closely monitored and evaluated using key metrics such as ROUGE-L to assess the quality of the model's output.

5. **Optimization:** To enhance model efficiency, Parameter-Efficient Fine-Tuning (PEFT) techniques were implemented.
6. **Report Generation:** The final step involved generating medical reports from the trained model and evaluating the quality of these reports to ensure they met clinical standards and provided accurate diagnoses.

Below is an in depth explanation of each of these stages.

My contributions:

1. Data Collection and cleaning:

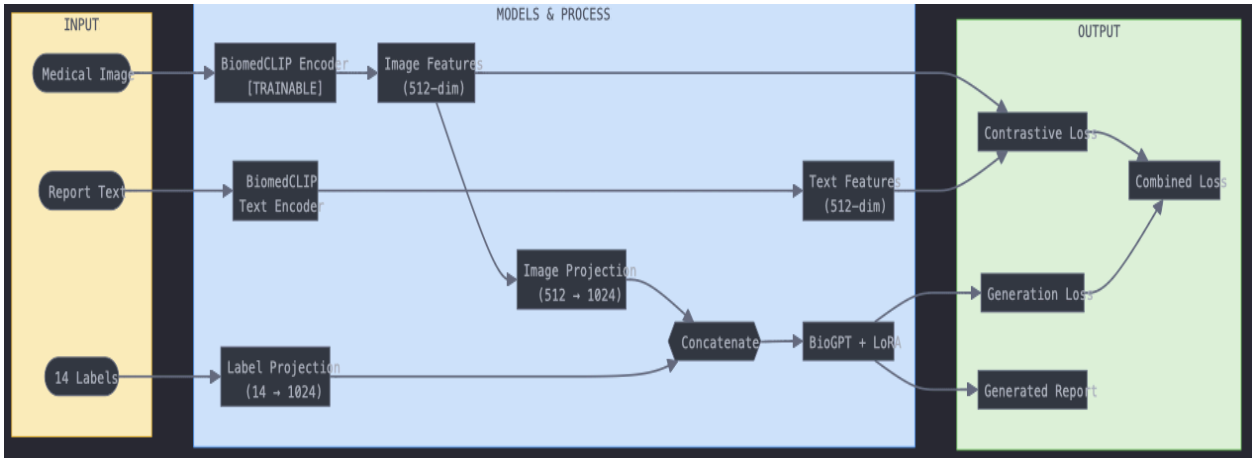
a. After working with the initial sample dataset from MIMIC, we wanted a bigger dataset to improve the model. So I downloaded the data from MIMIC using a shell script, running it continuously on a free t2 machine in AWS for 4 days to download nearly 120 GB of data. Wrote the script to compress the big PNG files to under 200kb.

2. Data Transfer:

a. After attempting to transfer data between our teammates' EC2 machines using SCP and other methods, I sought Professor Jafari's guidance. Following their recommendation, I created a Google bucket and used it to maintain all data and model transfers.

3. Built and trained the ChexNet model.

After experimenting with BioViLT, I found through several research papers that the model could be improved further by adding another layer of multi-label classification as input to the main ViLT models. Therefore, I decided to fine-tune ChexNet, a deep learning model based on the DenseNet-121 architecture, utilized for multi-label classification of chest X-ray images. It has been pretrained on chest X-rays, and its primary function was to identify structural abnormalities such as cardiomegaly, pneumonia, and atelectasis. The integration of ChexNet provided structured findings that enhanced the contextual understanding of the images, thereby improving the quality of generated medical reports.



The challenge was that the dataset was heavily imbalanced, so I had to use methods like a custom WeightedBCELoss class that applies different weights to positive and negative samples, uses WeightedRandomSampler to oversample minority classes, and implements inverse frequency weighting to give more importance to minority classes. We achieved an F1-micro score of 0.70.

4. Built and trained Biomed + ChexNet + BioGpt model.

After implementing models like BioViLT along with ChexNet, we decided to try out Biomed-clip with ChexNet and BioGPT because Biomed-clip is adapted for tasks requiring high generalization across unseen medical data. We initially used Biomed as a text embedding extractor, and then we wanted to determine whether training the entire model—along with image and text encoder with BioGPT—for report generation would improve performance, unlike BioViLT where we train only the alignment model and BioGPT. Following this analysis, I proceeded to train the complete model as well.

Biomed Training configuration :

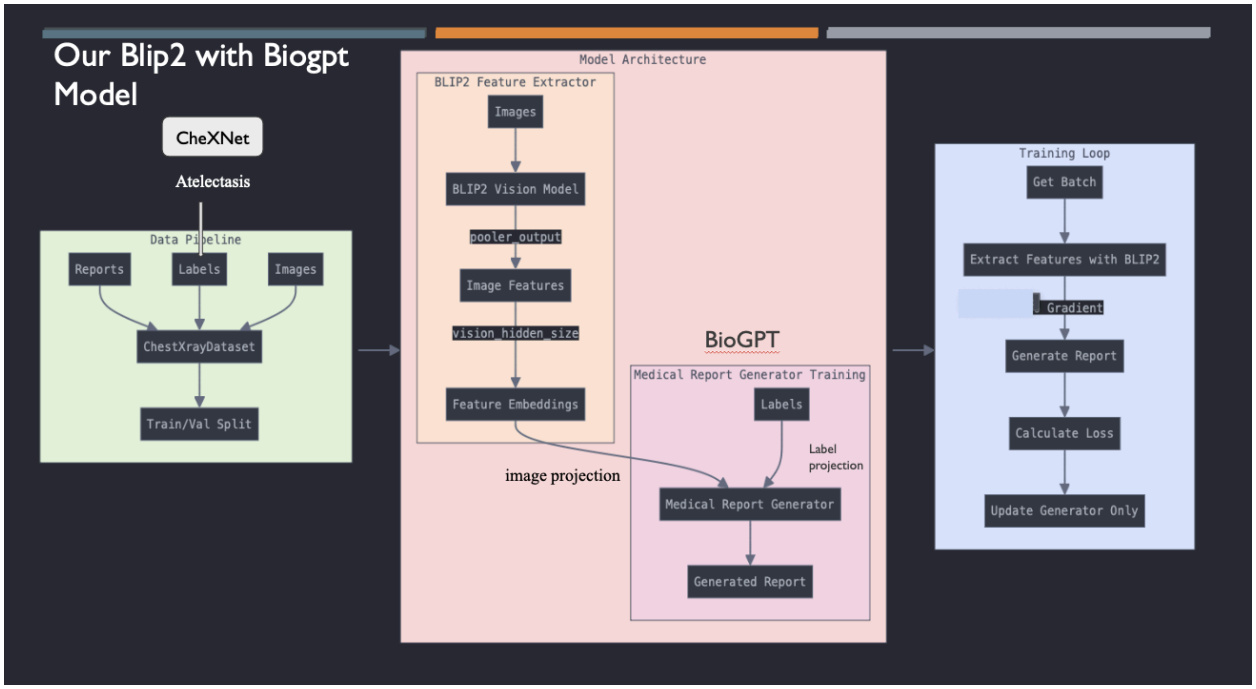
- **Training Setup**:
  - Batch size: 8
  - Number of epochs: 18
  - Training/validation split: 80/20
  - For BiomedCLIP model: AdamW with learning rate of 1e-5
  - Uses ReduceLROnPlateau scheduler for both optimizers
- Training Specifics:
  - Loss function: Combines contrastive loss and generation loss
    - 30% weight for CLIP contrastive loss
    - 70% weight for generation loss
  - Early stopping patience: 5 epochs
  - Uses ROUGE-L score for model selection

- **BioGPT training configuration**
  - **PEFT Configuration:** LoRA with
    - rank 16
    - alpha 32
    - dropout 0.1.
  - **Generation Parameters:**
    - max_length=150
    - temperature=0.38
    - top_k=50
    - top_p=0.585

After experimenting with these parameters, we found that these specific configurations performed well. Since Biomed was already trained on biological images, we wanted BioGPT to be more focused, so we kept the temperature, top_p, and top_k values small.

5. Built and trained Blip + ChexNet + BioGpt mod

From our experience with BioViLT and Biomed, we learned that training only alignment models, using Biomed as an image extractor, or even training the entire Biomed CLIP model were not sufficiently effective approaches. Therefore, I wanted to experiment with a larger model using an innovative approach. Through our research of academic papers, we discovered BLIP, a well-generalized model that uses an innovative QFormer which can be trained while keeping the image and language models frozen. However, I ultimately used it only as a text embedding extractor since I was unable to train it due to computation power limitations. In the end, I trained only the ChexNet + BioGPT component.



BioGPT training configuration

- **PEFT Configuration:** LoRA with
  - rank 16
  - alpha 32
  - dropout 0.1.
- **Generation Parameters:**
  - max_length=150
  - temperature=0.78
  - top_k=50
  - top_p=0.985

6. Create inference scripts for all the models developed.

In order to run the model I trained on Streamlit, I created inference scripts for all the models I built and trained - ChexNet, BLIP, and BiomeD models.

Conclusion:

Using ChexNet to add labels for the images as input to the models yielded significant benefits and improved overall performance. Although we trained the entire Biomed model, our best ROUGE score was still lower than that achieved with the BLIP model when used as a text embedding extractor. So training bigger models with a good amount of data will significantly improve the ROUGE L score.

FURTHER IMPROVEMENTS

1. Training the Q-former (Alignment module) of Blip 2 model to enable cross attention so that image embeddings and text embeddings are aligned.
2. Use an ensemble of a custom CLIP and BLIP model could also improve the performance.

Percentage of code copied from the internet around 20 to 25%

REFERENCES

1. https://huggingface.co/ChantalPellegrini/RaDialog-interactive-radiology-report-generation

2. BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs
3. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation
4. https://huggingface.co/docs/transformers/en/model_doc/blip-2
5. https://huggingface.co/microsoft/BiomedCLIP-PubMedBERT_256-vit_base_patch16_224
6. https://stanfordmlgroup.github.io/projects/chexnet/