Automated Medical Report Generation of Chest X-Ray Images Using Encoder-Decoder Architecture and Transfer Learning

Senthil Thanneermalai
Rice University
6100 Main St, Houston, TX 77005

Anupreet Sihra
Rice University
6100 Main St, Houston, TX 77005

Shambhavi Kurup
Rice University
6100 Main St, Houston, TX 77005

Kaushal Kumar Agarwal
Rice University
6100 Main St, Houston, TX 77005

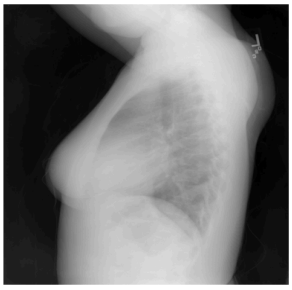
***Abstract*—Advancements in deep learning have opened new possibilities in medical imaging, particularly in automated report generation from diagnostic images. In this project, we build a deep learning-based system for the automatic generation of medical reports using chest X-ray images. We extracted image features through transfer learning, employing the CheXNet model tailored for pneumonia detection from Chest X-ray data. We began with a straightforward retrieval-based system and then progressed to encoder-decoder models employing simple RNNs. Additionally, we experimented with different decoder architectures such as LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) and observed that the models with GRU (Gated Recurrent Unit) networks performed better than the decoders in this setting. To further optimize our model, we attempted to integrate a visual attention mechanism, which did not perform as well as we expected. We suspect that this may be due to the complexity of the chest X-ray images, resource limitations, or potential mismatches between the model architecture.**

1. Background & Motivation

Chest X-rays in the medical field are used for diagnosing many problems such as cancer, pneumonia, different kinds of infections, heart-related lung problems, and more recently COVID-19. However, manually analyzing these X-rays is time-consuming and prone to errors. Using deep learning, we aim to automate the generation of medical reports that can mitigate these issues, thereby speeding up the diagnostic process and reducing the manual workload of radiologists. A groundbreaking study [1], made important strides by employing co-attention techniques and a hierarchical long short-term semantic memory (LSTM) to accurately analyze medical images, showing great promise in both radiology and pathology. More recent work [2] in 2022 indicates that the problem is still not fully solved and requires high attention to enhance diagnostic accuracy in the medical field. For such a system to be deployable, we require much higher accuracy. Some ideas for future work include modifying loss functions, experimenting with different datasets, and exploring various model architectures.

We have rigorously experimented with various deep learning models and techniques for interpreting chest X-rays, thereby attempting to achieve a high accuracy and efficiency in our diagnostic approach. For this project, we used the Indiana University Chest X-ray dataset, which includes 7470 images with frontal and lateral scans and associated radiology reports as shown in Figure 1. This project is a step towards a future where technology and healthcare work hand-in-hand for better patient care everywhere.

Frontal Scan



Lateral Scan

Original Report: *The cardiac silhouette and mediastinum size are within normal limits. There is no pulmonary edema. There is no focal consolidation. There is no XXXX of a pleural effusion. There is no evidence of pneumothorax.*

Predicted Report: *the cardiomediastinal silhouette is within normal limits for size and contour. The lungs are normally inflated without evidence of focal airspace disease, pleural effusion or pneumothora. no acute bone abnormality*

Figure 1. Sample data point from the IU X-Ray dataset, along with the reference report and our model's prediction

2. Methods and Experiments

We made no augmentations to the image data as our target variable (reports) is sensitive to any modifications in the X-rays. Commonly used image transformations such as rotations and flips would alter the report as well. Hence, we only performed image resizing and normalizing. For text data, we performed basic cleaning and expanded contractions. For this project, we have selected the Indiana University Chest X-ray images consisting of 7470 images and reports. After visualizing the dataset, we noticed that each patient has a minimum of 1 chest X-ray and a maximum of 4 chest X-rays (2 Frontal and 2 Lateral). Since most patients only had 1-2 associated scans, we retained 2 scans for all patients and concatenated image features of both before training. We replicated the first scan for patients having only one. After merging the two image features and dropping the null values, we had a dataset consisting of images from 3500 patients. We divided the dataset into training, testing, and validation images. The training dataset consisted of 3200 images whereas the test dataset and validation dataset consisted of 150 images each. As part of the data preprocessing phase, we removed extra spaces, changed it to lowercase, and expanded contractions.

2.1. Experiment 1 - Retrieval-Based Method

To serve as a baseline, we started with a simple retrieval-based model that uses a state-of-the-art CNN pre-trained on ChestX-ray14, which is currently the largest publicly available chest X-ray dataset, to extract features from the chest X-ray images. We used *k*NN to determine the image feature vector that is closest to the others and return its report. As expected, we received a low BLEU score mentioned in the Results section.

2.2. Experiment 2 - Encoder-Decoder Model

In the Encoder part of our model, we used CheXNet (a model pre-trained on Chest X-rays that has an accuracy of 95% in pneumonia detection) to extract medically relevant image features. CheXNet was specifically trained for pneumonia detection. As our problem statement does not involve pneumonia, we extract the feature representation from the second last layer instead of the classification layer. The Decoder portion of our model consisted of GRU Layers that were trained on our text data.

Through this process, the network is able to identify relationships and patterns in the text data. GRU (Gated Recurrent Unit) layers are employed because of their capacity to represent sequential data and identify textual dependencies. The Image layer and the Text layer were fused together and passed through a fully connected layer. This step is critical to create a unified representation consisting of both text data and image data. We incorporated Dropout layers in the architecture to prevent overfitting in the network by randomly deactivating a certain percentage of neurons in the layer. We experimented with dropout values ranging between 0.2 and 0.7, and observed that the model performed best when 50% of neurons were deactivated. In the Decoder section, we experimented with the following configurations:

1) Simple RNN - Basic Recurrent Neural Network that is trained on sequential data.
2) LSTM (Long Short Term Memory) - Capable of remembering long-range dependencies using a set of gates.
3) GRU (Gated Recurrent Network) - Similar to LSTM but it can be trained much quicker.
4) GRU + LSTM - Leverages both combinations for increased complexities.

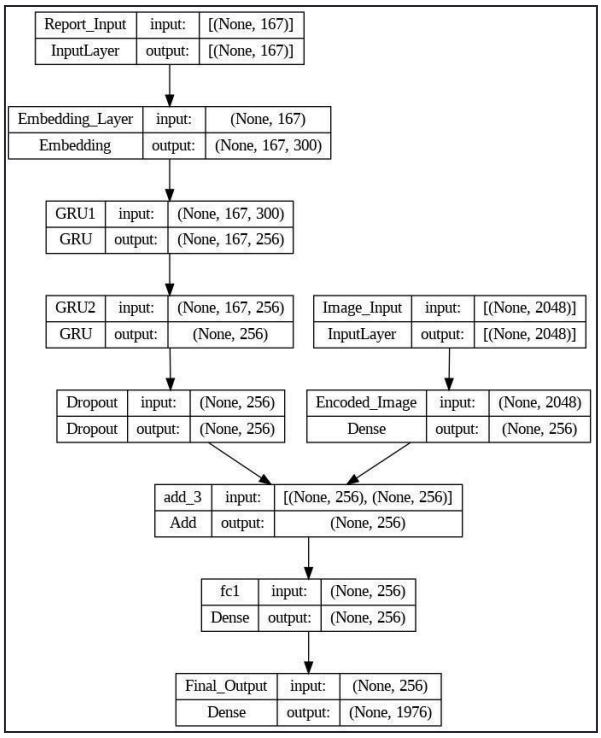


Figure 2. Encoder Decoder Model Architecture

2.3. Experiment 3 - Encoder-Decoder Model with Attention

We also attempted to enhance our model's performance by integrating the encoder-decoder model with a visual attention mechanism (Luong's type 1 attention mechanism, similar to the Bahdanau attention mechanism) to help it learn the relationship between parts of the X-ray and relevant parts in the report. We passed the image feature vectors (extracted from CheXNet) through the attention mechanism, which computes the attention weights by comparing the decoder's hidden state with the encoder's output states. The attention weights are then used to create a context vector, representing the relevant regions in the image. We implemented the decoder as an LSTM network with an additional attention layer.

3. Results

Figure 3 shows the BLEU scores for the models we experimented with.

Model	BLEU1	BLEU2	BLEU3	BLEU4
Retrieval Based Model	0.25	0.1482	0.0979	0.0695
RNN (as Decoder)	0.3069	0.1833	0.1071	0.0503
GRU + GRU (as Decoder)	0.322	0.1948	0.1231	0.0681
GRU + LSTM (as Decoder)	0.3203	0.1875	0.1071	0.0452
Attention Based Model	0.1169	0.0182	0.0019	0.0008

Figure 3. BLEU scores for all models

We observed that the predictions of our final model (with the GRU decoders) consisted mostly of the high-frequency words present in the dataset. An overwhelming majority of the model's predictions on the test set were one of the following reports: *"the lungs are clear bilaterally. specifically no evidence of focal consolidation pneumothora or pleural effusion. the cardio mediastinal silhouette is unremarkable. visualized osseous structures of the thora are without acute abnormality."* or *"the lungs are clear. there is no pleural effusion or pneumothora. The heart and mediastinum are normal. the skeletal structures are normal"*. So, the predicted reports demonstrated a significant skew toward normal patient outcomes.

We evaluated the quality of the predicted reports using the BLEU (Bilingual Evaluation Understudy) score, a widely used metric in deep learning used to evaluate the quality of texts generated in natural language processing tasks. The BLEU score consists of a brevity penalty and measuring n-gram overlap. The brevity penalty is the component that penalizes generated translations that are too short compared to the ground truth text, to discourage the model from producing overly concise outputs to maximize precision. The n-gram overlap component is what measures how many n-grams in the generated text overlap with n-grams in the reference text. So, the BLEU1, BLEU2, BLEU3, and BLEU4 scores in Figure 3 correspond to 1-grams, 2-grams, 3-grams, and 4-grams. BLEU scores range from 0 to 1, with a score of 0 indicating no overlap between the generated text and the ground truth and a score of 1 indicating perfect overlap between the two. A strong correlation between human evaluations of translation quality and BLEU scores has been

shown. A BLEU score of 0.5-0.6 is considered to indicate generated texts of very high quality.

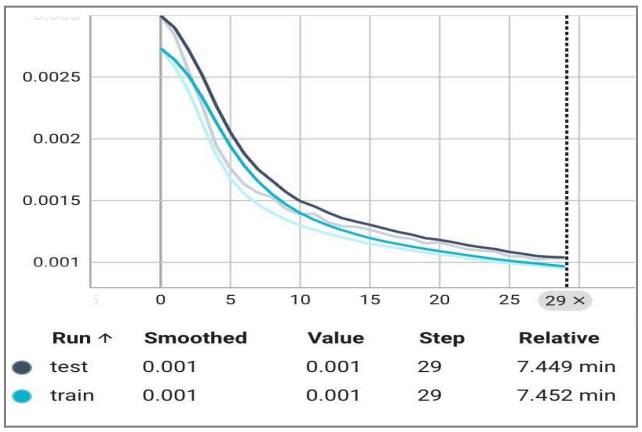


Figure 4. Loss Plot for Encoder-Decoder Model (using GRU)

Our naive *k*NN approach gave a BLEU1 score of 0.25. Our Simple RNN model took an average training time of 125 seconds per epoch using a Tesla T4 GPU and gave a BLEU1 score of 0.3069. Our LSTM model performed slightly better with a BLEU score of 0.31 in approximately the same time. Our model with a GRU layer and an LSTM layer in the decoder part of the model gave better performance with a BLEU score of 0.322. After multiple experiments, we concluded that using two GRU layers in the decoder in the model gave the best performance, although only marginally better, with an average training time of 15 seconds per epoch on T4 and a BLEU score of 0.322.

Our model with the incorporated attention mechanism didn't perform as well as we expected. We got lower BLEU scores than the Encoder-Decoder model as mentioned in Figure 3, even though we did see a decrease in both the training and validation loss. We think this might have happened due to several reasons that we didn't get to explore due to time constraints. We discuss possible ways to improve the model in the next section.

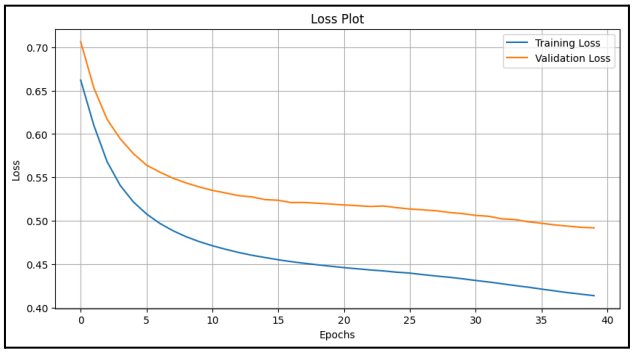


Figure 5. Loss Plot for Encoder-Decoder Model with Attention Mechanism

4. Discussion & Conclusion

Our deep learning models generated reports correspond to more normal outcomes than abnormal ones because of the inherent skew in the dataset. Most patients have normal diagnoses. The consistent "normal" patterns in predicted reports suggest our model learned from a narrow perspective. Due to the medical nature of the dataset, we cannot artificially generate more image-report pairs through data augmentation to balance the skew in the dataset. The only way to deal with this issue is to use a different Chest X-ray dataset that has more balanced patient outcomes.

We observed through our experiments that using GRU as a text-decoder in our model architecture achieved a higher BLEU score (of 0.322), compared to Simple RNNs and LSTM. We think that relying solely on Encoder-Decoder models (RNN, GRU, LSTM) failed to capture the complexity of the data. To boost localized information processing, we also integrated a visual attention mechanism in our Encoder-Decoder model that we hoped would help the model focus on relevant parts of the image during report generation. However, with limited time constraints, we were only able to achieve a BLEU score (of 0.117). It is worth exploring different hyperparameters, such as the number of attention units, hidden state vector dimensionality, etc. We did tune these hyperparameters to a certain extent but were constrained by available computational resources. Additionally, we also think it is worth exploring other attention mechanisms (hard vs. soft attention, global vs. local attention, etc.). It is also worth noting that our attention results might be due to an implementation flaw on our part. Going forward, we hope to improve our model performance by implementing these changes.

5. Code Availability

The link to our code repository is as follows: https://github.com/kaushalag29/COMP576-Final-Project

6. Group Member Roles

Anupreet: Worked on Bahdanau attention mechanism to improve normal encoder-decoder performance and hyperparameter tuning

Kaushal: Worked on the extraction of image features using a pre-trained CheXNet model and performed experimentation with distinguished neural networks (RNN, LSTM, GRU) to optimize the accuracy of the Encoder-Decoder model

Senthil: Worked on text preprocessing, building the Encoder-Decoder architecture, helper functions required for

the model to train on the dataset and HyperTuning parameters.

Shambhavi: Worked on building the Encoder-Decoder model with Attention Mechanism. Performed hyperparameter tuning, and experimented with 2 attention mechanisms - Luong type 1 and Luong type 2.

References

[1] B. Jing, P. Xie, and E. P. Xing, On the Automatic Generation of Medical Imaging Reports, https://arxiv.org/pdf/1711.08195.pdf

[2] E. Darici, A. Timashov, M. Tan, and K. Yu, "Chest X-Ray Report Generation from Chest-X Ray Images Stanford CS224N Custom Project.". https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1224/reports/custom_117157386.pdf

[3] R. Kumar Soni, "Indiana University-chest X-rays automated report generation," Medium, https://rohansoni-jssaten2019.medium.com/indiana-university-chest-x-rays-automated-report-generation-38f928e6bfc2

[4] S. Qureshi, "Chest X-Ray Medical Report Generation Using Deep Learning," Analytics Vidhya, Feb. 04, 2021. https://medium.com/analytics-vidhya/chest-x-ray-medical-report-generation-using-deep-learning-bf39cc487b88