**Capstone Project – Automatic Chest X-Ray report generation**

|  |  |
| --- | --- |
| **Ezhirko Arulmozhi** | [ezhirko.arulmozhi@gmail.com](mailto:ezhirko.arulmozhi@gmail.com) |
| **Kalyan Aravapalli** | [kalyan.aravapalli@gmail.com](mailto:kalyan.aravapalli@gmail.com) |
| **Kunal Majumdar** | [kunalmajumdar123@gmail.com](mailto:kunalmajumdar123@gmail.com) |
| **Naga Pavan Kumar Kalepu** | [nagapavan.kalepu@gmail.com](mailto:nagapavan.kalepu@gmail.com) |

**Abstract**

The automatic generation of radiology reports given medical radiographs has signiﬁcant potential to operationally and improve clinical patient care. Although deep learning techniques have been successfully applied to image classification and image captioning tasks, radiology report generation remains challenging in regard to understanding and linking complicated medical visual contents with accurate natural language descriptions. We propose a new encoder-decoder model to deal with the modern-day challenges. In this model, we extract the key features from the image using a pre-trained model (CheXnet) and pass it to the LSTM network. We also use BioWordVec/BioSentVec embeddings that are created on the PubMed and MIMIC-III datasets. In addition, to enrich the decoder with descriptive semantics and enforce the correctness of the deterministic medical-related contents such as mentions of organs or diagnoses, we extract medical concepts based on the radiology reports in the training data and fine-tune the encoder to extract the most frequent medical concepts from the x-ray images. Such concepts are fused with each decoding step by a word-level attention model. The experimental results conducted on the Indiana University Chest X-Ray dataset demonstrate that the proposed model achieves the state-of-the-art performance compared with other baseline approaches.

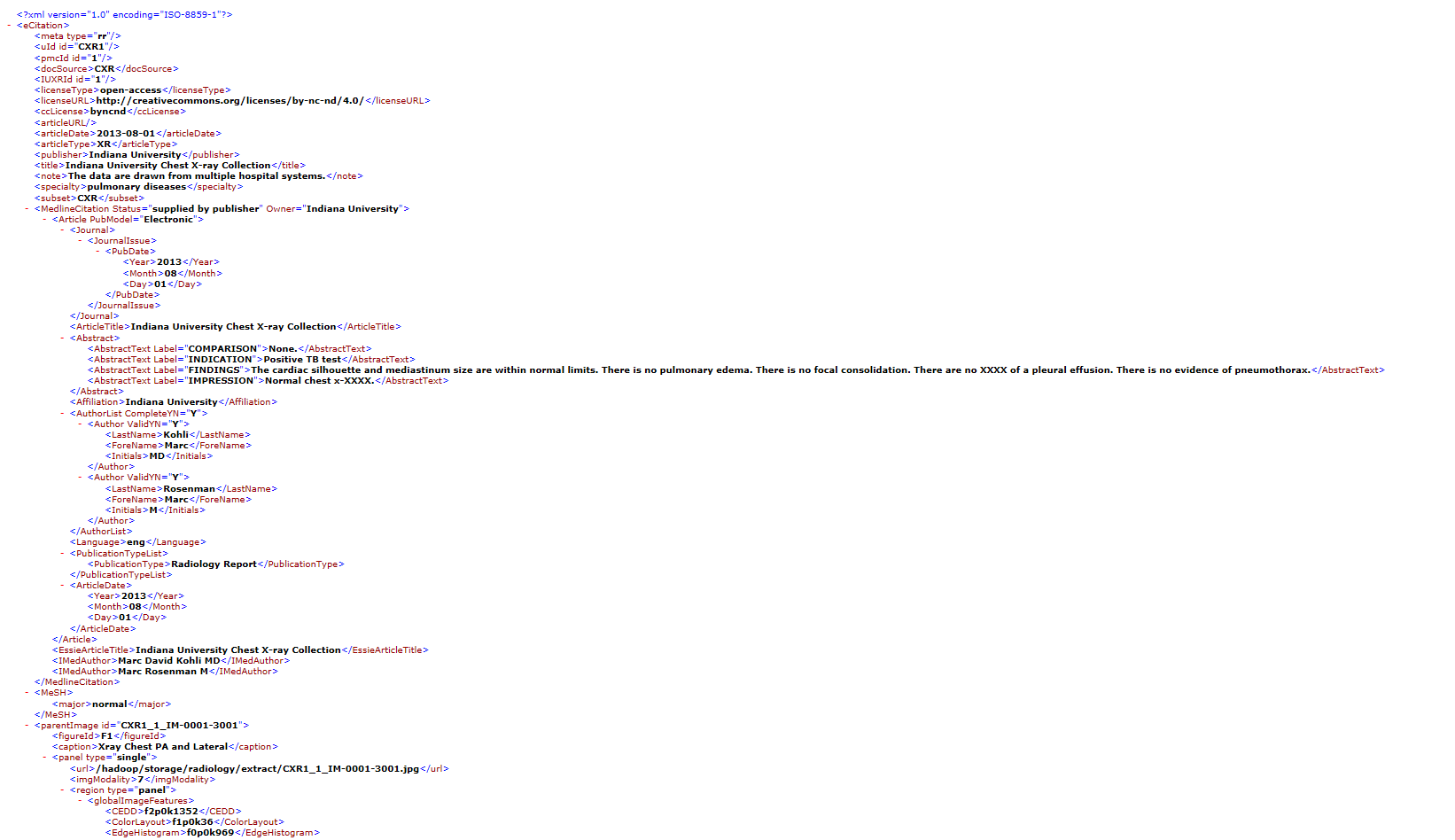
**Data Sneak-Peak**

Data for this work is provided in it’s raw format by the Indiana University for non-commercial use. It’s provided in the following format:

NLMCXR\_png --> Folder with 7k images

NLMCXR\_reports --> Folder with 3.5k reports

**Fig1: Sample image: Fig2: Sample Report:**

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwib4ZWYmJHlAhXvzoUKHRDUAT8QjRx6BAgBEAQ&url=https://lhncbc.nlm.nih.gov/system/files/pub9175.pdf&psig=AOvVaw2I9aX3TSmGLjmCjqV9vKD6&ust=1570779603122235) 

**Exploratory Data Analysis**

The following analysis was carried out as part of the EDA activity

1. Total number of reports equals to the total number of people
2. Each report can have multiple X-Ray images (each identified by the attribute **figureId** within the XML file. Values include F1, F2, F3...,)
3. X-Ray images include both Frontal and Lateral images as shown below. Both images uniquely identified by the attribute parentImage id in the report XML file.

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwib4ZWYmJHlAhXvzoUKHRDUAT8QjRx6BAgBEAQ&url=https://lhncbc.nlm.nih.gov/system/files/pub9175.pdf&psig=AOvVaw2I9aX3TSmGLjmCjqV9vKD6&ust=1570779603122235) 

**Fig3: Frontal and Lateral images**

1. Quite a lot of reports either don’t have lateral images or have multiple lateral images associated with one person.
2. The following attributes are found in the reports that can be considered as features in generating the report:
   1. COMPARISON --> Not much information available, hence can be ignored
   2. INDICATION --> Forehand information (Key feature)
   3. FINDINGS --> Report summary (Key Feature)
   4. IMPRESSION --> Report supplementary (Key feature)
3. It’s identified that almost 530 reports have missing Findings which is a key factor in generating the report.
4. Close to 28 reports don’t have either FINDINGS or Impression.
5. Apart from the regular special characters, strings like '<BR>', '&gt;]', '[&lt;', '?' were also found in the data
6. Identified data inconsistency where some statements were starting immediately after a period while the others were rightly starting after a period followed by a space

**Pre-Processing**

Based on the EDA carried above, the following Pre-Processing steps were carried out on the data

1. Used BeautifulSoup to extract the following key attributes information from the XML files:
   1. parentImage id --> Provided mapping between report and image
   2. figureid --> Provides information if the image is a Frontal or Lateral. F1 --> Frontal, anything other than F1 is a lateral.
   3. label: --> Key features for report generation
      1. FINDINGS
      2. IMPRESSION
      3. INDICATION
2. For the 530 reports identified in the EDA that don’t have FINDINGS, treated IMPRESSION as FINDINGS
3. The 28 reports and their corresponding images that don’t have any key information within the reports were ignored from training
4. Data was cleaned up using the following strategy
   1. Converted the text to lower case
   2. Removed punctuations except period (.) which was required to generate sentences separated by a period.
   3. Added a space where the statements were starting immediately after a period to make the data consistent across
   4. Removed special characters and special strings like '<BR>', '&gt;]', '[&lt;', '?' which were part of the data and identified as part of EDA
   5. Didn’t remove the numerical values as they form a key portion in the report.
5. Generated Word Vocabulary from the entire corpus
6. Also generated Sentence Vocabulary from the corpus
7. Added two tokens in every paragraph as follows:
   1. 'startseq' -> This is a start sequence token that's added at the start of every paragraph.
   2. 'endseq' -> This is an end sequence token that's added at the end of every paragraph.

**Hybrid Model Architecture**

The components of model architecture are

1. CNN model to encode images
2. NLP Embedding model to embed the word/sentence into numerical vectors
3. LSTM model to capture the context and temporality of words in a paragraph to generate the report syntactically and semantically.
4. Sampling/Report generation technique
   1. Greedy Search
   2. Beam Search

**CNN Model to encode images:**

To encode images we have used pre-trained model CheXNet which is a 121-layer convolutional neural network trained on over 112,120 frontal-view X-ray images of 30,805 unique patients with 14 diseases. CheXNet intended to predict 14 pathology diseases. The base model for CheXNet is DenseNet-121,

* Before inputting the images into the network, we need to downscale the images to **224×224** and **normalise** based on the **mean and standard deviation** of images in the **ImageNet training set.**
* We capture the output of last but one layer by removing the last softmax layer and the output vector size is 1024.

**NLP Embedding model to embed the word/sentence into numerical vectors:**

**<Need to fill this content by Kunal>**

**LSTM model to capture the context**

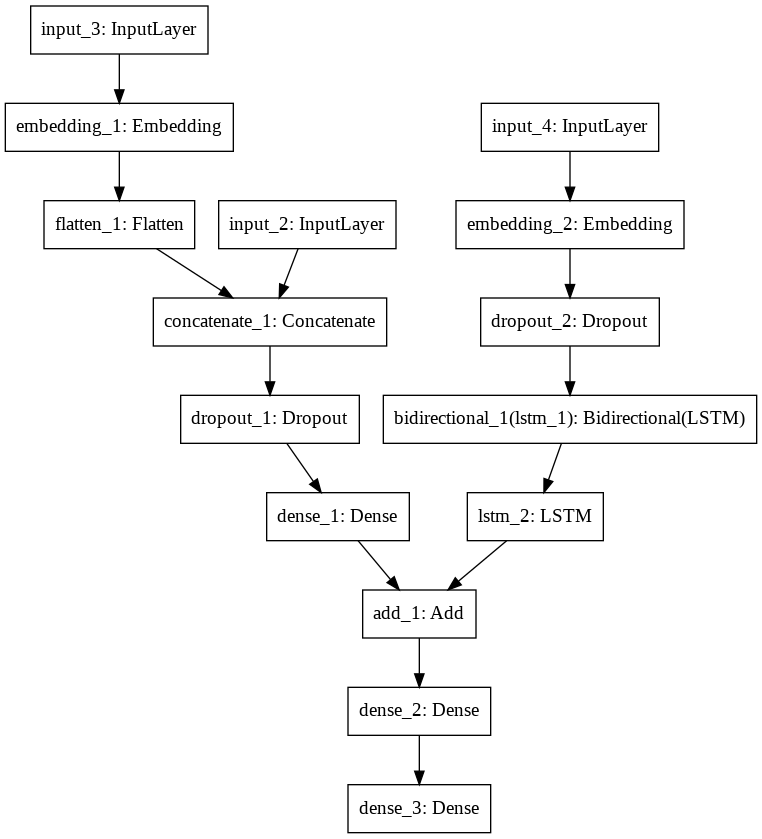
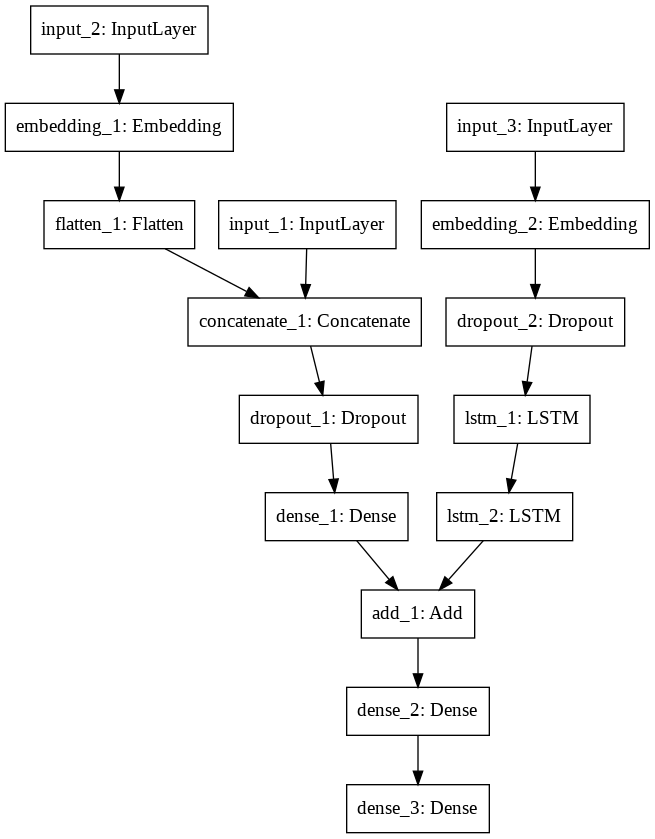
We tried with two different LSTM architectures

1. LSTM with Attention LSTM layer
2. Bidirectional LSTM

Above both models significantly well. We tried with 256 and 500 LSTM cells. We increased the number of cells to generate longer paragraphs. LSTM layer takes the input from Embedding layer which are numerical vectors representations of word/sentence. Here input size is 200 dimensional vector in case of word based model and input size is 700 dimensional vector in case of sentence based model.

It is encoder-decoder architecture, encoder is LSTM layer/s and decoder takes the input from activations of dense layers connected to Image encodings which are generated with the help of CNN pre-trained model CheXNet-121 & activations of LSTM layer & cell states of LSTM layer in case of attention model. It is easier to comprehend through below pictorial architecture.

**Fig4: Attention model architecture** **Fig5: Bidirectional model architecture**



**Sampling/Report generation technique**

**Beam Search:**

Any of the above model architectures generate probability scores all words in Vocabulary we created but that won’t automatically generate report. So we need a sampling algorithm to do this task. Beam search is one of the best sampling algorithms to generate sentence or paragraphs.

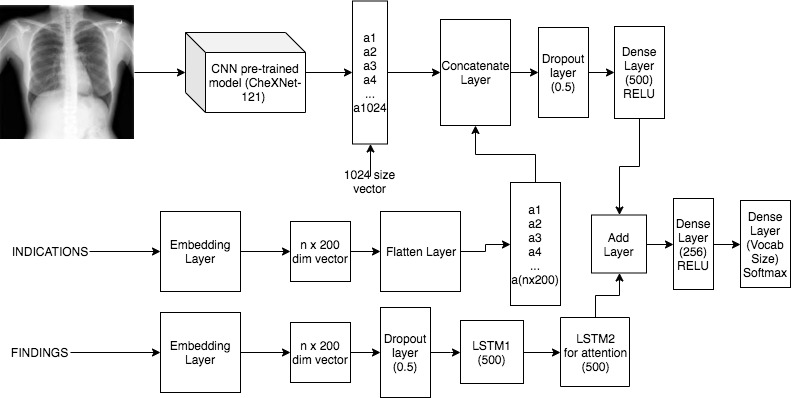
**How Beam Search works if Beam search width = 3:**

* When we feed start word (we have defined startseq) to either of above models, model will predict the probabilities for all words in vocabulary.
* Capture the words with top 3 highest probabilities.
* Append these 3 words individually to startseq and feed to above model. And it again predicts the probabilities for all words in vocabulary.
* Here is key concept of Beam search that it will do calculate conditional probability by multiplying previous word probability with all current probability scores of vocab words. And it will capture the top 3 conditional probabilities.
* And above process repeats till the end word get predicted (we have defined endseq as end word).

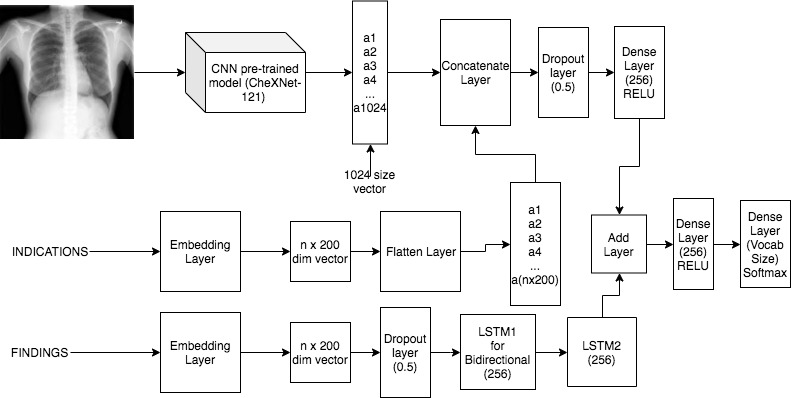
There is another sampling technique called Greedy search which is nothing but Beam Search with Width =1. Beam Search with more width generates better sentences but computationally expensive.

**End to End flow of Model in detail:**

**Attention Model:**

****

**Bidirectional Model:**

****