AUTOMATED TEXT GENERATION FROM RADIOLOGY IMAGES PROJECT REPORT | JANUARY - JUNE 2022

Dekai Zhang
Imperial College London
Department of Computing
dz819@ic.ac.uk

Dan Schofield
NHS England
Transformation Directorate
daniel.schofield1@nhs.net

ABSTRACT

Automated medical image interpretation is still only reaching a low amount of its potential opportunity. While computer vision research has shown promising results in dealing with a high degree of variation in medical images (e.g., different machines having different angles and contrasts), this has largely been focused on classification tasks. This differs in format to the outputs from human medical image interpretation which are typically captured in free-text and are not confined to a potentially limiting set of labels.

This work was completed as part of an NHS England Data Science PhD Internship project. The work described in this report aims to explore advances in multi-modal machine learning that take advantage of the relationship between images and text, enabling a shift in focus from classification to free-text as output.

1 Introduction

Advances in machine learning have resulted in algorithms that can perform automated image or text recognition tasks at such a level that they rival human performance. Their success has resulted in increasing interest in their application to healthcare, where these algorithms could speed up workflows and reduce medical errors. The automation of chest x-ray interpretation has been one area of particular interest given the prevalence of this type of scan [1]. This project contributes towards this area by aiming to generate radiology reports from chest x-ray images.

There are two main challenges that this project addresses. First, while a number of works have demonstrated that machine learning models can yield strong performances in classifying chest x-ray images [2, 3], there remains a question of suitability of these outputs. Specifically, healthcare practitioners are more likely to be used to consuming free-text reports rather than classification labels for a fixed number of pre-determined classes as commonly produced by most machine learning models [4].

A second, related challenge lies in the common requirement of most machine learning methods for carefully labelled data [5]. For chest x-rays, this can imply a prohibitive demand for radiologists to perform the labelling before model training can even commence. This is compounded by patient privacy concerns which pose an obstacle to collecting the volume of data necessary to meet the demands of most machine learning models.

This project set out to explore machine learning methods that would be capable of producing free-text reports from images, thus addressing the first challenge. Our efforts focus on a recent class of machine learning methods [6] that do not only meet this primary challenge but are also less vulnerable to the second challenge. Intuitively, these methods aim to learn the relationship between images and text by contrasting matching and non-matching image-text pairs. In doing so, they are able to digest existing radiology studies (comprising images accompanied by a free-text report) and train models that are able to find a description that matches an image and vice versa. Our exploration results in `TxtRayAlign` – an application of this contrastive method to the MIMIC-CXR radiology dataset.

In the following, Section 2 provides an overview of some of the main approaches that have been considered in this project. Section 3 presents the MIMIC-CXR radiology dataset used for training and evaluating our model. Section 4 defines `TxtRayAlign` and evaluation criteria. Section 5 contains technical details on implementation. Section 6 details

the main experiments with results given in Section 7. Section 8 discusses limitations and potential future directions enabled by this work and Section 9 concludes.

2 Background

In this section, we will first provide an overview of the main approaches with which radiology reports can be generated from images (Section 2.1) as well as other approaches which generate text from images but which may differ substantially from radiology reports in format and content (Section 2.2).

2.1 Report generation

Within the current literature two principal approaches have emerged for radiology report generation. In the first (Section 2.1.1), the problem is framed as an image captioning task, whereby text is generated from scratch. In the second approach (Section 2.1.2), which we take, the problem is re-framed as a retrieval task, whereby text is instead retrieved from an existing corpus.

2.1.1 Image captioning

In the previous literature, one body of work has viewed report generation as an image captioning task for which encoder-decoder structures have been successfully deployed in the non-medical domain [7]. Figure 1 illustrates such an encoder-decoder structure for chest x-ray interpretation: an image encoder first embeds the image in some latent space before this latent representation is then used by a text decoder to reconstruct this representation as text.

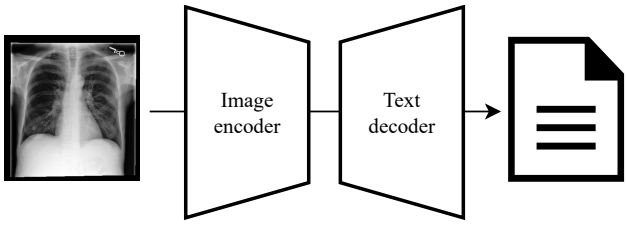


Figure 1: An encoder-decoder structure for text generation. A query image is encoded. The latent representation of the image is then passed into a text decoder which returns the reconstruction of the image in text space as output.

Variations on this general structure with different architectures for the image and text encoder have been proposed for generating radiology reports [8–11]. Boag et al. [4], however, draw out the distinction that this type of approach may excel more at generating readable natural language text rather than clinically relevant or even clinically accurate text. They further demonstrate that simple retrieval-based approaches, which we will outline in the next section, perform better in generating text that contains accurate clinical content.

2.1.2 Report retrieval

In the second approach, report generation is viewed as a retrieval task, whereby images are submitted and reports are generated out of pieces of text retrieved from a large corpus of reference reports. This has the advantage that the generated reports are guaranteed to be readable (i.e., fulfilling natural language criteria) while containing clinically relevant information, thus also raising the likelihood of clinical accuracy.

Boag et al. [4], for instance, encode the images associated to the reports and then perform a nearest-neighbour search over these encoded images to retrieve the report of the closest image. This approach is, however, purely based on image similarity and therefore neglects to exploit the relationship between the images and the text.

Zhang et al. [5] propose a contrastively trained pair of image and text encoders which learns to align the representation of an image with that of its associated report. This alignment between the image and text representations allows for the retrieval space to encompass both. With this in hand, reports can be generated for an image by (i) computing the embeddings of both the image in question as well as the corpus of reference reports and (ii) findings the nearest reports of that image in the shared latent space (Figure 2).

Endo et al. [12] apply this approach to the image-to-text retrieval task and use a pair of image and text encoders contrastively pre-trained on 400 million pairs of natural images and text [6]. They demonstrate that this encoder pair

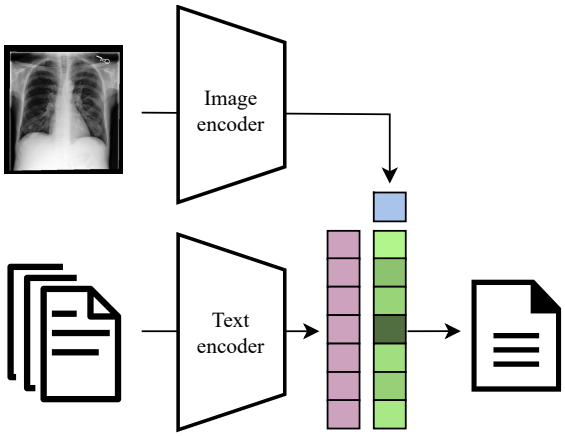


Figure 2: A contrastive retrieval mechanism. A query image is encoded and compared with the embeddings of a corpus of reference reports. The report with the greatest cosine similarity in the shared embedding space is returned as the output.

outperforms earlier image captioning models in clinical accuracy. The contrastive approach provides a number of additional advantages:

1. The retrieval mechanism is symmetric between the different data modalities and can be similarly applied for text-to-image and image-to-image retrieval tasks as Zhang et al. [5] do.
2. Training on pairs of images and text matches more closely how radiology studies are being produced as part of existing clinical workflows, thus avoiding otherwise prohibitively labour-intensive labelling
3. The image and text encoders can be initialised from encoders pre-trained on more common classification tasks, allowing re-deployment of already familiar architectures.
4. Contrastively trained models generalise well on a range of classification tasks [6, 13, 14].

This project aims to explore and evaluate the suitability of this approach for generating radiology reports from images. Points of particular interest include its feasibility given a more modestly sized radiology dataset (approximately 0.1% of the natural image-text dataset used by Radford et al. [6] and approximately 0.01% of the one used by Zhai et al. [14]) and limited hardware resources. Zhang et al. [5] and Endo et al. [12], who use the same dataset as in this project, suggest that data size will not necessarily be a binding constraint in successfully training a contrastive model but both use large encoders that may not train well in a low resource settings. In this project, we therefore investigate and compare the performance of contrastive models using smaller encoders on the basis of Ke et al. [2] who demonstrate that smaller image encoders can draw even in classification performance with larger ones.

2.2 Other approaches

In this section, we provide a brief overview of other image-to-text approaches which have have been considered but were not taken forward, as their output differs from radiology reports in both format and content. Nevertheless, these approaches could provide complementary perspectives.

2.2.1 Explanation methods

The focus of this project is to generate text that addresses the challenge of providing interpretable outputs to clinical practitioners. Explanation methods address an orthogonal challenge – that of explaining the inner workings of oftentimes "black-box" machine learning models generating those outputs. This area of research is growing quickly and encompasses a great variety in how explanations are generated, what these explanations relate to and what format they take. [1]

Lee et al. [16] provide one approach to generating textual and visual explanations for an image classifier trained to detect malignant breast masses. Figure 3 reproduces these explanations and shows that the visual component consists of a heat map highlighting the most salient areas while the textual component provides a short justification. While this

[1] See Arya et al. [15] for a recent survey.

approach generates text from an image, it is unclear how well its application to chest x-rays would be able to capture the format and content of radiology reports which may be more descriptive than justifying in nature. Furthermore, it is not clear how well this approach generalises to chest x-rays which are generally much larger images containing a correspondingly greater variety in tissue types and areas.

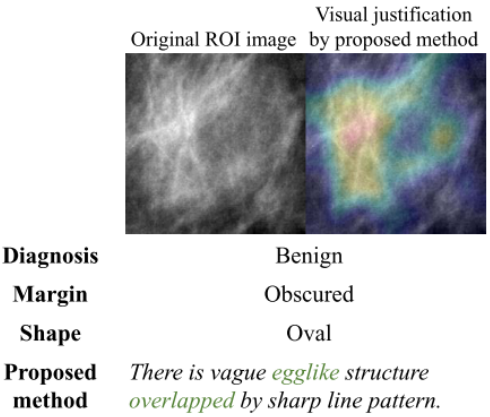


Figure 3: An example textual and visual explanation for an image classification (reproduced from [16]).

Explanations can take other shapes. Silva et al. [17], for instance, have focused on finding "similar" images, or prototypes, which may help clinical practitioners compare and contrast with other relevant cases (illustrated in Figure 4). Notably, this type of explanation could be provided using the contrastive retrieval approach we opt for, as the model can be repurposed for image-to-image tasks.

Figure 4: Four prototypes with the same (green) and different (red) classification labels as the test image (reproduced from [17]).

2.2.2 Visual question answering

Visual question answering (VQA) systems enable users to ask questions about an image in natural language and receive answers in natural language (Figure 5). This approach differs from explanation methods, as the answers are not necessarily informative about the workings of the model. Moreover, VQA systems are trained specifically for this task and may differ from explanation methods, many of which are applied to models that have been trained on classification or other tasks. This again poses the same challenge as before: training a VQA system requires a carefully curated dataset consisting of questions and answers which may be difficult to produce [18, 19].

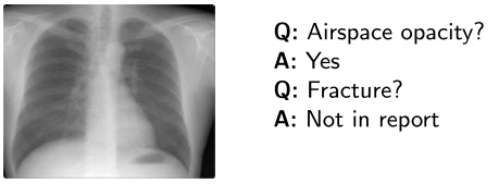
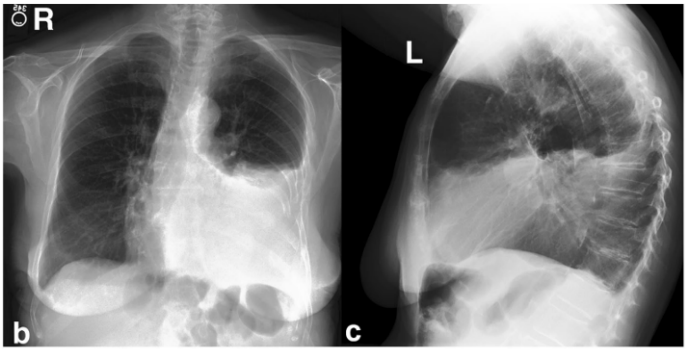


Figure 5: An example VQA interaction (reproduced from [18]).

3 Dataset

For all of our experiments, we are using the MIMIC-CXR dataset [20, 21], which consists of 227,835 de-identified radiology studies containing 377,110 chest x-ray images for a population of 65,739 patients. We separately describe the images (Section 3.1) and reports (Section 3.2). This dataset was recorded in the United States at the Beth Israel Deaconess Medical Center Emergency Department between 2011 and 2016. Figure 6 shows an example study.



EXAMINATION: CHEST (PA AND LAT)

INDICATION: ___ year old woman with ?pleural effusion // ?pleural effusion

TECHNIQUE: Chest PA and lateral

COMPARISON: ___

FINDINGS:

Cardiac size cannot be evaluated. Large left pleural effusion is new. Small right effusion is new. The upper lungs are clear. Right lower lobe opacities are better seen in prior CT. There is no pneumothorax. There are mild degenerative changes in the thoracic spine

IMPRESSION:

Large left pleural effusion

Figure 6: An example radiology study contained in MIMIC-CXR (reproduced from Johnson et al. [20]).

Section 3.1 describes the image part while Section 3.2 describes the text part of the dataset.

3.1 Images

The images were originally made available in DICOM format. Johnson et al. [21] provides MIMIC-CXR-JPG – a compressed version of the dataset in which every image has been converted into the JPEG format and normalised to have pixel values in the range of [0,255].

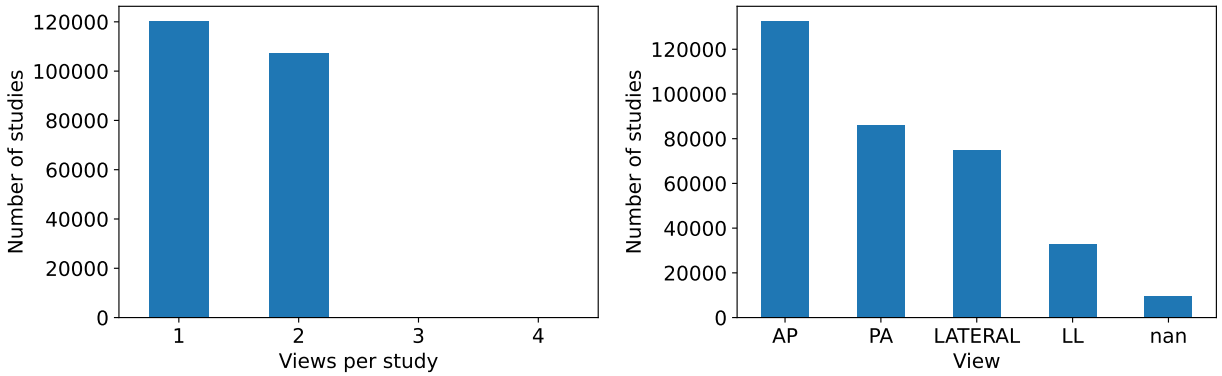
The original DICOM version takes up approximately 4.6TB of storage space while the JPEG version takes up approximately 557GB. While the compression results in a loss in information, which may negatively affect training, JPEG format images are widely used in computer vision. In the following, we use the JPEG images from MIMIC-CXR-JPG for all of our experiments for ease of integration.

The images have been taken from different views with approximately two thirds taken either from the front or the rear ("AP" and "PA" in Figure 7b) and the remainder taken from the side ("LATERAL" and "LL" in Figure 7b). Almost half of all studies contain more than a single view (Figure 7a). 16% of studies have images taken from the same view, which correspond to images taken from the same view but rotated or re-captured using different imaging settings. Since the dataset does not make clear which image is the original and which are duplicates, we have not removed any.

3.2 Reports

Each study contains a number of chest x-ray images paired with a single free-text report. The report itself typically contains a number of sections where the "Findings" and "Impression" sections (see Figure 6) corresponds to those that the radiologist writes to describe the images. For studies with multiple images, the report typically does not explicitly refer to individual images separately, meaning that the report cannot be easily split and assigned to each image.

MIMIC-CXR provides a table with multi-label, multi-class classifications for the last section of each report using the CheXpert labeller [22], which determines if a given report mentions a label and, if so, whether the mention corresponds to a positive, negative or unclear finding.



(a) Number of different views per study.

(b) Frequency of different views.

Figure 7: Images have been taken from different views. Almost half of the studies have more than one view.

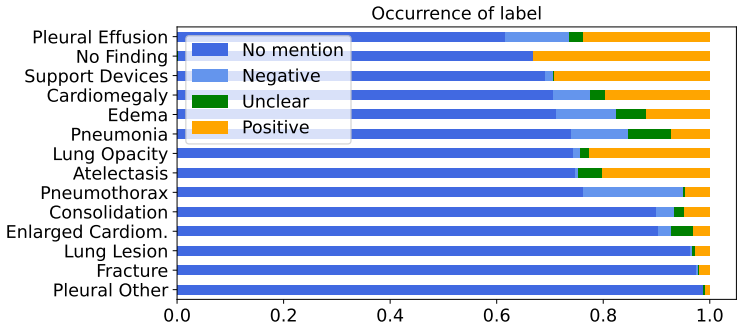


Figure 8: Frequency of labels from CheXpert classification of reports.

Figure 8 presents an overview of the frequencies with which each of the labels occur, suggesting a dataset that is imbalanced in both labels and classes: (i) there are labels that occur much more rarely than others (e.g., "Pleural Other" vs "Pleural Effusion") and (ii) for some labels, some classes occur much more frequently (e.g., Positive for "Support Devices" vs Negative for "Pneumothorax"). These imbalances are particularly noteworthy in the context of a retrieval task, as they not only potentially impact training performance but also place limitations on the retrieval space.

Since reports may contain more than a single label, an additional source of data imbalance arises in the co-occurrence of labels. Figure 9 maps out their correlations and suggests that there are some labels which occur together more frequently than other pairs (e.g., "Atelectasis" with "Pleural Effusion" as opposed to with "Pneumonia"). These correlations may not be as problematic, as reports can be split into constituent sentences, which should help to reduce correlations present on the report-level.

4 Methodology

In this section, we first describe the contrastive method underlying `TxtRayAlign` (Section 4.1) and then detail the criteria we use to evaluate the performance of the model (Section 4.2).

4.1 Contrastive training

Following the works of Radford et al. [6], Zhai et al. [14] and Zhang et al. [5] we use a model with a two-encoder structure, one for text and one for images, as shown in Figure 10. During training, a batch of N image-text pairs is taken. Each of the images and texts are separately encoded. The text and image embeddings, z_i^T and z_i^I, are then compared for their cosine similarity by taking their dot product. The objective of the training procedure is to maximise the cosine similarity of the N matching image-text pairs and minimise that of the remaining $N^2 - N$ non-matching pairs. For a given batch, the cosine similarities are then used to calculate the probability that a given pair is a match, which can be defined as:

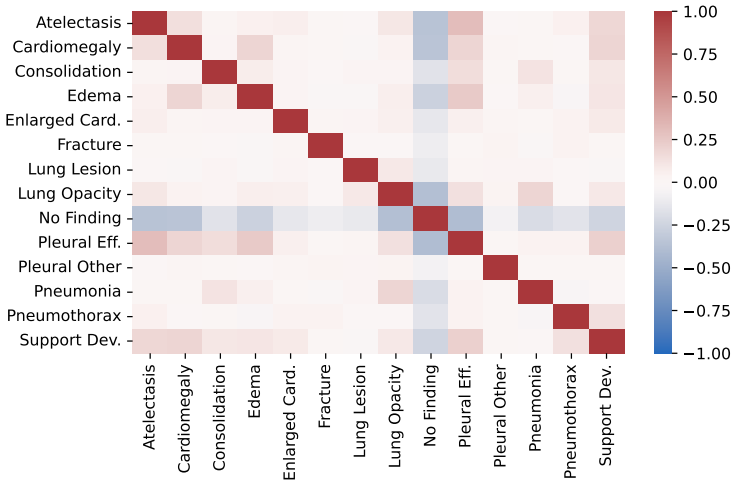


Figure 9: Correlation map between CheXpert labels contained in reports.

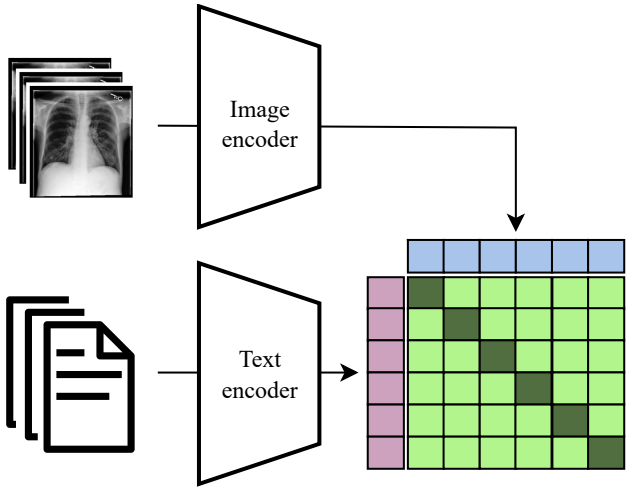


Figure 10: The contrastive training regime aims to maximise the cosine similarity of the embeddings of matching images and text (on the diagonal) and minimise the cosine similarity of non-matching pairs (on the off-diagonal).

$$P(z_i^I, z_i^T; \tau) = \frac{exp(z_i^I \cdot z_j^T / \tau)}{\sum_{i=0}^{N} exp(z_i^I \cdot z_k^T / \tau)} \tag{1}$$

where τ is a trainable temperature parameter. This results in the InfoNCE loss which symmetrically measures the success in maximising the similarity of matches and minimising the similarity of non-matches, defined as:

$$L_{\text{InfoNCE}} = -\frac{1}{2}\left[\frac{1}{N}\sum_{i=0}^{N} logP(z_i^I, z_i^T; \tau) + \frac{1}{N}\sum_{i=0}^{N} logP(z_i^T, z_i^I; \tau)\right] \tag{2}$$

Cheng et al. [23] argue that this loss results in pairs being viewed as hard labels which is not desirable when a given image could be matched with multiple pieces of text in the batch and vice versa. This is a salient point for training on radiology studies where a given report could be associated with more than one image and where a given image could be described with more than a single piece of text. To incorporate such "soft" matches, the authors propose to keep an exponential moving average (EMA) of the model during training to use as a teacher for self-distillation (as in [24]), which consists of an averaging between the weights of the previous teacher model θ'_{t-1} and the current student model θ_t:

$$\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t \tag{3}$$

where $\alpha = 0.999$. Given the teacher, they add an additional loss which measures the Kullback-Leibler (KL) divergence between the probability distribution of matches over a given batch, which they define as:

$$L_{KL} = \frac{1}{2}\left[KL(P_M^I, P_{EMA}^I) + KL(P_M^T, P_{EMA}^T)\right] \tag{4}$$

where P_M^I is the match probability (Equation 1) distribution across the images in a given batch calculated by the current model and P_{EMA}^I the same but calculated by the EMA teacher. Symmetric definitions follow for the match probability distribution across the pieces of text. This results in an overall loss – which we adopt for `TxtRayAlign` – defined as:

$$L = L_{\text{InfoNCE}} + L_{KL} \tag{5}$$

4.2 Evaluation criteria

Boag et al. [4] show that commonly used natural language generation metrics, such as BLEU or CIDEr, do not capture the quality or accuracy of generated clinical text very well. One approach to address this shortcoming has been to rely on classifications of the generated and ground truth reports and using classification metrics as a proxy for content accuracy. In the context of radiology studies, we follow Boag et al. [4] who use the CheXpert report labeller [22] which is a deterministic, rule-based model for classifying radiology reports. Other options exist – Smit et al. [25] propose a BERT-based labeller which was used by Endo et al. [12] and could increase evaluation accuracy. Figure 11 illustrates the evaluation framework.

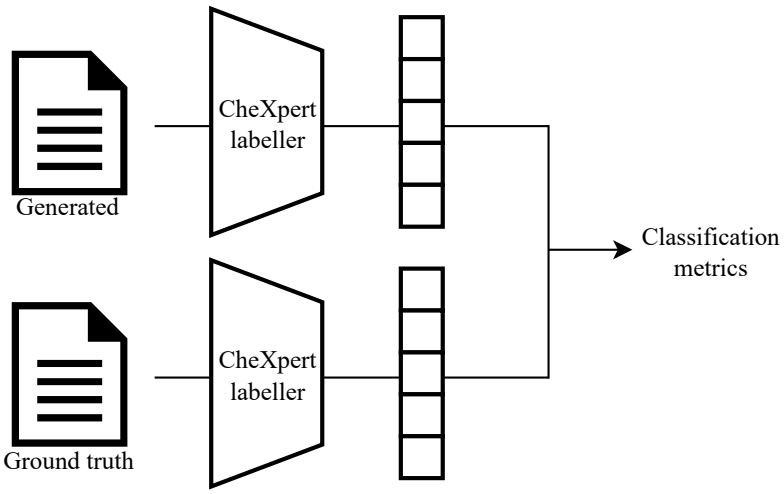


Figure 11: The content of the generated and ground truth reports are classified with the CheXpert labeller [22] and then compared to derive classification metrics.

For the classification metrics, we adopt flat-hit@k, precision@k, recall@k and F_1@k. Intuitively, a flat-hit is recorded when for a given test instance i with labels L_i there is an overlap in labels with any of the k retrieved pieces of text $\{\hat{y}\}_k$. Formally, flat-hit@k is defined as:

$$\text{Flat-hit@k} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}\left(\{\{\hat{y_i}\}_k \cap L_i\} \neq \emptyset\right) \tag{6}$$

Precision@k measures the proportion of relevant findings in the retrieved text – high precision thus reflects few irrelevant findings in the retrieval. This can be formally defined as:

$$\text{Precision@k} = \frac{1}{N}\sum_{i=1}^{N} \frac{|\{\{\hat{y_i}\}_k \cap L_i\}|}{|\{\hat{y_i}\}_k|} \tag{7}$$

Recall@k measures the proportion of findings covered by the retrieved text – high recall thus reflects exhaustive coverage of findings for the test instance. This can be formally defined as:

$$\text{Recall@k} = \frac{1}{N} \sum_{i=1}^{N} \frac{|\{\{\hat{y}_i\}_k \cap L_i\}|}{|L_i|} \tag{8}$$

F_1@k is a harmonic mean of precision and recall, and therefore represents a summary of both. It is defined as:

$$F_1\text{@k} = 2 \, \frac{\text{Precision@k} \times \text{Recall@k}}{\text{Precision@k} + \text{Recall@k}} \tag{9}$$

5 Implementation details

In this section, we discuss implementation details (Section 5.1), list the hardware we used (Section 5.2) and provide an overview of other related codebases (Section 5.3).

5.1 Pre-processing

The JPEG images in MIMIC-CXR have a relatively high resolution in the area of 1500 x 2000 pixels which we proportionally downscale to images with a length of 256 pixels on their shortest side. This is done for two reasons: (i) training with batches of images in their original size is likely costly in terms of memory consumption and read speed, and (ii) most commonly used image encoders have an optimal input size which implies worse performance for images either too small or too large [26]. The optimal input size for the image encoders we use in our experiments is 224 x 224 pixels. We arrive at this size by further applying a random crop to the typically rectangular images.

Chest x-rays are taken in black and white, resulting in a single greyscale channel, whereas most image encoders assume colour images with three RGB channels. We convert the single-channel black and white images into a compatible format by replicating the greyscale channel value across each of the three RGB channels.

We normalise all of the images using the commonly used mean and standard deviation from ImageNet. We note that there could be an incremental benefit from calculating the mean and standard deviation for MIMIC-CXR and applying those for the normalisation.

As described in Section 3, any given study may contain more than one image taken from different views, so that the report potentially describes findings only seen in some of the images. For simplicity, we treat each image independently and pair it with the report of the study it is part of. This implies that there will be images which will be paired with reports describing findings that the image does not contain, potentially increasing the noisiness of the dataset. Future improvements could include multi-view encoding of the images.

For the text, we found that pairing each image with the entire report during training proved too memory-consuming. Instead, we split each report by sentence and randomly select one of the resulting sentences for every training batch. Given a large enough number of batches, every sentence of every report should appear. With text compression, chunking, or greater memory, pairings could be extended to include multiple sentences, if not the entire report.

5.2 Hardware overview

The training was performed on two machines hosted by Microsoft Azure with the following main specifications: (i) 1 Nvidia Tesla T4 GPU, 4 vCPUs from an AMD EPYC 7V12 processor, running Windows Server 2019 and (ii) 4 Nvidia Tesla T4 GPUs, 64 vCPUs from an AMD EPYC 7V12 processor, running Ubuntu 18.04. Inference was performed on the first machine only.

5.3 Codebase

The training script for `TxtRayAlign` is largely based on the `train-CLIP` repository by Gordon [27] which is a PyTorch Lightning implementation for contrastively training image and text encoders, either initialised with individually chosen encoders or with contrastively trained encoder pairs provided by Radford et al. [6].

At the start of this project, there were a number of codebases which have been considered and are listed in Table 1. We chose not to adopt these, as they were either designed to focus on classification tasks, or limited to training a narrow range of models (typically the ones shown in [6]), or they did not seem to have users who reported successful training runs, or were only fully published over the course of the project. It is important to note that some of these projects were still in development as of writing this report.

Table 1: Related codebases.

Name	URL
CLIP	`https://github.com/openai/CLIP`
OpenCLIP	`https://github.com/mlfoundations/open_clip`
WiSE-FT	`https://github.com/mlfoundations/wise-ft`
OTTER	`https://github.com/facebookresearch/OTTER`
x-clip	`https://github.com/lucidrains/x-clip`
ConVIRT	`https://github.com/edreisMD/ConVIRT-pytorch`

6 Experiments

In this section, we provide an overview of the training setup (Section 6.1), hyperparameters (Section 6.2) and evaluation setup (Section 6.3).

6.1 Training setup

We train the model with different encoder pairings on 1%, 5% and 100% of the data for 50 epochs[2] – a summary of the different experiments is provided in Table 2. For each dataset size, we use a 90-5-5 split between training, validation and test subsets, saving the checkpoint of the model with the lowest validation loss. For the image encoder, we use ResNet50 and EfficientNetB0 – both of which have been pre-trained on ImageNet. ResNet50 is a commonly used architecture for chest x-ray classification. EfficientNetB0 is less frequently used and smaller than ResNet50 but has been shown to perform competitively in chest x-ray classification if pre-trained on ImageNet [2].

For the text encoder, we use DeCLUTR, BioclinicalBERT and DistilBERT. The first two have been pre-trained on scientific papers and clinical notes respectively and can be considered in-domain encoders. DistilBERT is a smaller version of BERT which has not been pre-trained on in-domain data.

Lastly, we also include the ResNet50-version of the CLIP model provided by Radford et al. [6]. This model has been contrastively pre-trained on images and text scraped from the internet which likely differ from medical images and notes.

The out-of-domain encoders likely require transfer learning before they can extract useful features for medical images and notes. Experiments with frozen encoders, as suggested in [14], did not appear to train well given the domain shift. Most of the pairings are also composed from individually pre-trained encoders which do not output embeddings into the same space, with the exception of the CLIP model whose encoders have already been contrastively aligned.

Training runs on 1% and 5% of the data respectively take approximately 1 hour and 3 hours on the single-GPU machine, whereas training runs on the full dataset take about 25 hours on the four-GPU machine (specifications in Section 5.2). Given resource and time constraints, only a select number of encoder pairings have been trained on the full dataset.

Table 2: Overview of experiments for each encoder pairing.

Encoder		Pre-training		Model size	Trained on		
Image	Text	Contrastive	In-domain	Parameters (m)	1%	5%	100%
EfficientNetB0	DistilBERT	No	No / No	71.7	✓	✓	
EfficientNetB0	BioclinicalBERT	No	No / Yes	113.6	✓	✓	
EfficientNetB0	DeCLUTR	No	No / Yes	115.2	✓	✓	✓
ResNet50	DistilBERT	No	No / No	91.9	✓	✓	
ResNet50	BioclinicalBERT	No	No / Yes	133.8	✓	✓	(✓)
ResNet50	DeCLUTR	No	No / Yes	135.4	✓	✓	
CLIP-ResNet50	CLIP-Transformer	Yes	No / No	102.0	✓	✓	✓

6.2 Hyperparameter choices

Table 3 reports the main hyperparameter choices. Notably, the batch size has direct bearing on the difficulty of the contrastive objective the models are trained with (Equation 2). Intuitively, for any given image, the model needs to

[2]With the exception of ResNet50-BioclinicalBERT on 100% of the data which was aborted after 32 epochs.

make the correct match out of the N reports in the batch. The probability of successfully finding the correct match decreases with the batch size. We set the batch size to 32 on the single-GPU machine following Zhang et al. [5], who found that slightly smaller or larger batch sizes (16 and 128) hurt performance. In contrast, Radford et al. [6] found that a very large batch size of 32,768 accelerated training. Since the four-GPU machine is able to accommodate larger batches, we ran the 100% data cases with a batch size of 512.

The image and text encoders that have not been jointly and contrastively pre-trained may output embeddings of different dimensions. It is therefore necessary to ensure common embedding dimensions. We achieve this by appending linear layers to both the image and text encoder which map the native embeddings into a common embedding space of the same dimensionality. We choose 768 as a common size used for text encoder embeddings. Preliminary experiments with 512 for the embedding dimension did not result in an improvement but more extensive experiments would be desirable.

During training, we use half precision floating point numbers to reduce the footprint. We also use the AdamW optimiser with a learning rate of 1×10^{-4} and a cosine annealing schedule with 1 warm-up epoch. Small experiments on 1% of the dataset with SGD as optimiser as well as smaller and larger learning rates did not yield an improvement in training behaviour.

Table 3: Overview of hyperparameters.

Parameter	Choice
Epochs	50
Batch size	32 for 1% and 5%, 512 for 100%
Embedding dimension	768 (for models other than CLIP)
Training precision	Half precision
Learning rate	1×10^{-4}
Optimiser	AdamW
Scheduler	Cosine annealing with 1 warm-up epoch

6.3 Evaluation setup

For the image-to-text retrieval task, we chose to retrieve $k = 2$ sentences to form the report which equals the choice in [12]. For the corpus of sentences, we obtain a sample of 11,522 sentences from the training and validation split of the full dataset. For the query images, we obtain a sample of 50 images from the test split of the full dataset. An overview of these choices is provided in Table 4.

To prevent any data leakage for models trained on a different subset of the data, we removed any training and validation images of any of the subsets under evaluation from the test split of the full dataset. To reduce calculation time, we pre-compute the embeddings of the query set and the corpus and save them to disk. A single evaluation run starting from pre-computed embeddings takes about 6 minutes.

Table 4: Overview of evaluation parameters.

Parameter	Choice
k	2
Number of query images	50
Number of corpus sentences	11,522

7 Results

In this section, we present the results from our experiments. We first discuss the training behaviour (Section 7.1) before providing an overview of how well the model is able to perform on the downstream image-to-text retrieval task (Section 7.2).

7.1 Training behaviour

Figure 12 and Figure 13 present the training accuracy and validation loss from training on 1% and 5% of the data respectively.[3] Note that the training accuracy is measured by the proportion of correct matches within a given batch and is therefore inversely proportional to the batch size.

The training behaviour on these two subsets of the data show clear similarities in training evolution between models using the same image encoder. The training accuracy appears to suggest that the CLIP model did best, followed by the models with ResNet50 and then by the models with EfficientNetB0 as image encoder.

The validation loss shows that the CLIP model reaches its optimum earlier than the other models, albeit at a higher validation loss, before starting to overfit. The ResNet50 group of model present the greatest decrease in validation loss, while the EfficientNet50 models only show a moderate decrease. Neither of the latter two indicate any overfitting. This difference between the CLIP model and the other individually initialised encoder pairings could come from various sources. Interestingly, the CLIP model uses the ResNet50 architecture for its image encoder, which may at first glance suggest it should behave similarly to the other models in the ResNet50 group. It is worthwhile noting that the pre-training for the CLIP model and the ResNet50 models differs in two respects: (i) the ResNet50 models were initialised with a ResNet50 image encoder pre-trained on ImageNet, a labelled image dataset, while CLIP was trained on relatively noisy image-text pairs scraped from the internet, and (ii) the ResNet50 encoder of the CLIP model has already been contrastively aligned to its paired text encoder, which may simplify training.

The training behaviour on the full dataset (Figure 14) differs in some respects. While the CLIP model still presents the greatest increase in training accuracy, its gain in accuracy over a no-skill model is proportionally much greater than on the smaller subsets. Note that with a batch size of 512, a no-skill model would have a chance of $\frac{1}{512}$ of being accurate. In contrast, the ResNet50 and EfficientNetB0 cases seem to be approximately equal in accuracy, even though their relative accuracy gains (of about 4x) are roughly similar to what can be observed on the smaller subsets.

The validation loss on the full dataset indicates some parallels with the smaller subsets: the CLIP model still tends to reach an optimum earlier before overfitting, while the other encoder pairings converge more slowly without overfitting. In contrast, the CLIP model reaches a significantly lower validation loss, followed by the EfficientNetB0 case and then the ResNet50 case this time. The difference in the training behaviour on the full dataset and the subsets is not entirely clear. One potential explanation could be related to the encoders used by CLIP which were contrastively pre-aligned on very large batch sizes. The CLIP model could therefore be better adapted to the step up in batch size used to train on the full dataset. The apparently worse performance of the ResNet50 and EfficientNetB0 cases could therefore be due to a suboptimal choice of hyperparameters. It could be worthwhile re-running these cases with smaller batch sizes as recommended by Zhang et al. [5].

7.2 Image-to-text retrieval evaluation

We compare how well the different models perform in image-to-text retrieval in Table 5 for models trained on 5% of the data and Table 6 for models trained on the full dataset.

Table 5: Image-to-text retrieval evaluation across 50 test instances. Models trained on 5% of the data.

Encoder						
Image	Text	Duplicates	Flat-hit@2	Precis.@2	Recall@2	F_1@2
EfficientNetB0	DistilBERT	8%	48%	22.2%	21.6%	21.9%
EfficientNetB0	BioclinicalBERT	14%	48%	23.2%	23.0%	23.1%
EfficientNetB0	DeCLUTR	8%	50%	20.4%	22.5%	21.4%
ResNet50	DistilBERT	8%	56%	25.3%	27.7%	26.5%
ResNet50	BioclinicalBERT	15%	56%	24.3%	27.4%	25.7%
ResNet50	**DeCLUTR**	**8%**	**62%**	**32.1%**	**29.1%**	**30.5%**
CLIP-ResNet50	CLIP-Transformer	15%	54%	25.0%	26.4%	25.7%

The performance of the models on the smaller subset are mostly consistent with their training behaviour described in the previous section. The ResNet50 family of models is performing best across all metrics. The CLIP model ranks second followed by the EfficientNetB0 models.

Interestingly, the models do not appear to improve significantly when trained on the full dataset: the CLIP model appears to benefit moderately (but is still worse than the ResNet50-DeCLUTR model trained on 5%), while the ResNet50

[3] Training losses for the 5% case are reported in Appendix A.

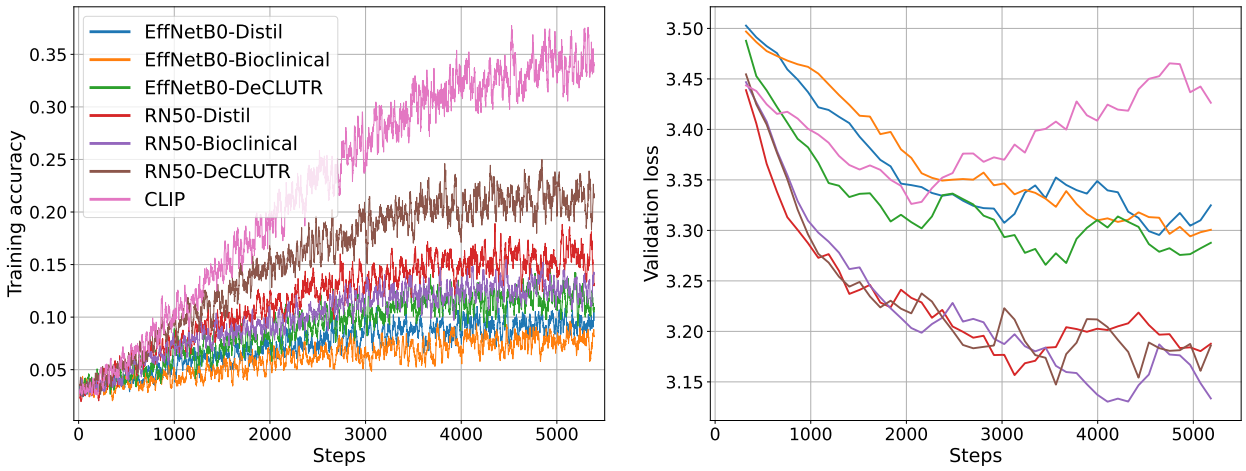


Figure 12: Smoothed training accuracy and validation loss on 1% of the data with a batch size of 32.

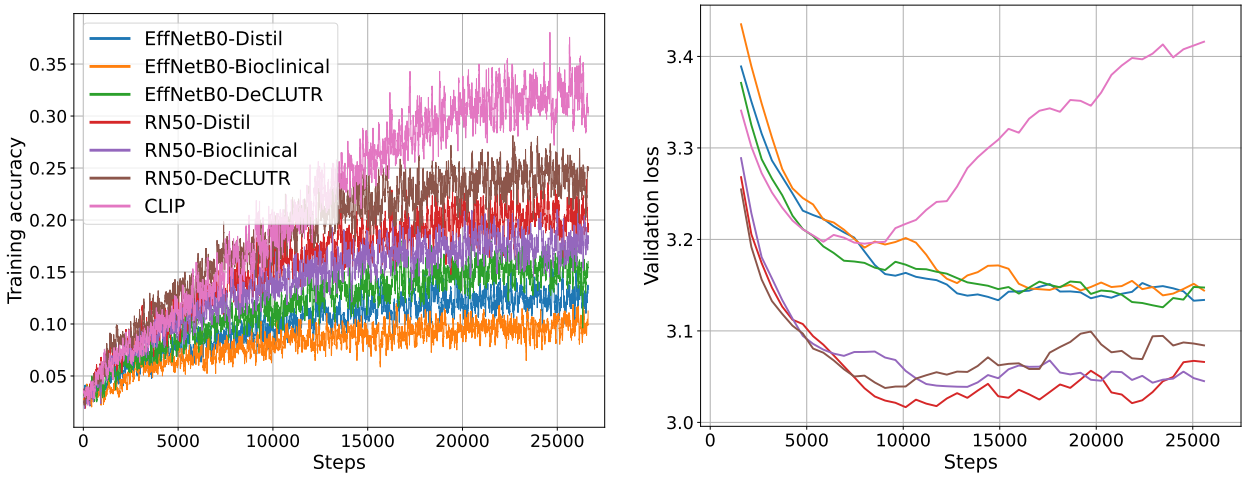


Figure 13: Smoothed training accuracy and validation loss on 5% of the data with a batch size of 32.

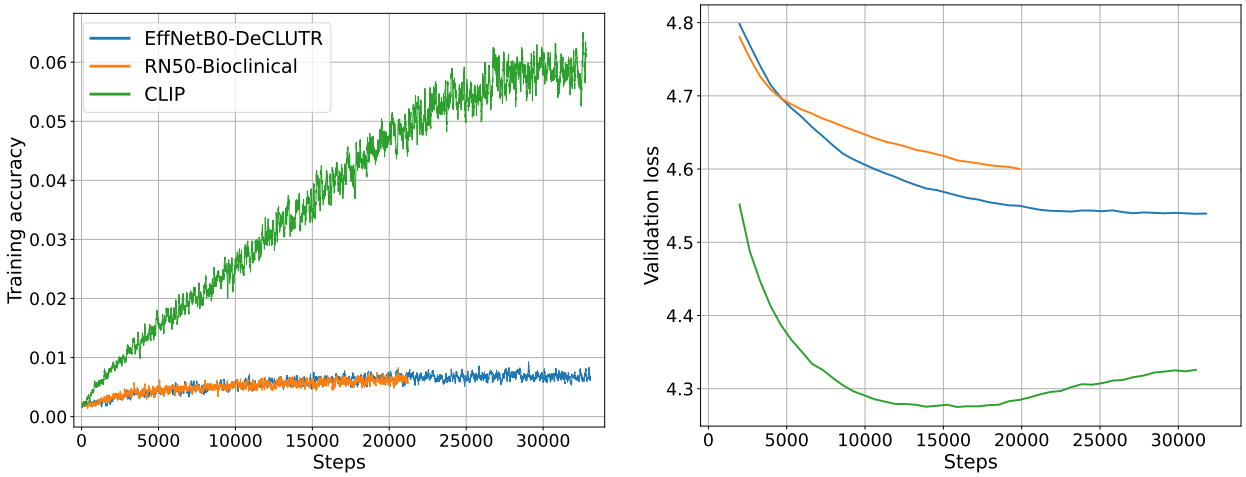


Figure 14: Smoothed training accuracy and validation loss on 100% of the data with a batch size of 512.

Table 6: Image-to-text retrieval evaluation across 50 test instances. Models trained on 100% of the data.

Encoder						
Image	Text	Duplicates	Flat-hit@2	Precis.@2	Recall@2	F_1@2
EfficientNetB0	DeCLUTR	3%	52%	20.7%	25.7%	22.9%
ResNet50	BioclinicalBERT	11%	44%	21.0%	23.2%	22.0%
CLIP-ResNet50	**CLIP-Transformer**	**12%**	**60%**	**27.0%**	**28.1%**	**27.6%**

and EfficientNetB0 models seem to deteriorate in performance. As discussed in the previous section, this could be an indication of suboptimal hyperparameters for this particular set of models.

In all cases, we observe that even the best performing model (ResNet50-DeCLUTR) only retrieves anything of relevance for 62% of queries. The retrieved sentences tend to contain findings that are not relevant for the query, as indicated by the relatively poor precision. Further, the query image contains findings that are only poorly covered by the retrieved sentences, as indicated by the low recall. These metrics are largely in line with those obtained by Endo et al. [12] and Boag et al. [4].

Figure 15 illustrates some of the strengths and failings of the model qualitatively. While the retrieved sentences appear to contain reasonably grammatical sentences and clinical language, they fail to accurately capture the severity of some of the findings (e.g., "moderately severe" vs "mild" edema) or causal relations between findings (e.g., "increased opacity [...] *due* to atelectasis"). Consistent with the quantitative measures, the retrieved sentence contain irrelevant information, as indicated by any sentences that are not highlighted on the right, and fall short in capturing all relevant findings, as indicated by any sentences that are not highlighted on the left.

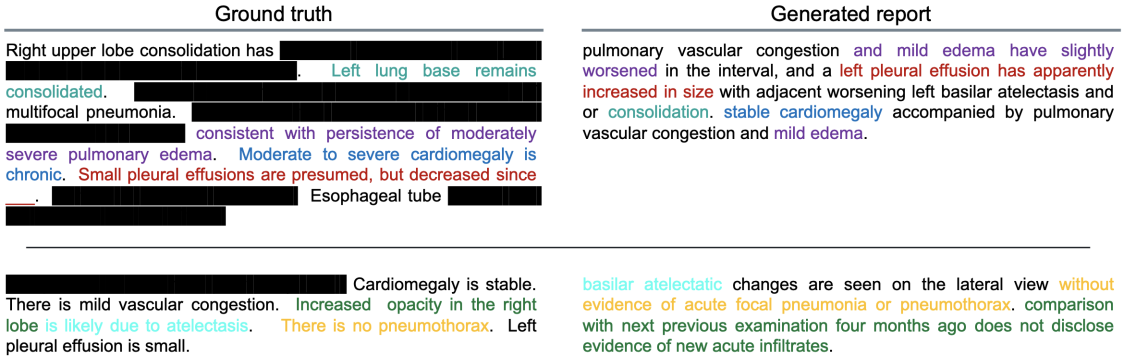


Figure 15: Two example reports generated by ResNet50-DeCLUTR (trained on 5%). Highlighted text corresponds to matches of the CheXpert sentence label between the ground truth and generated report. Ground truth report partially redacted for privacy.

8 Limitations and future directions

The work described in this report could be extended in a number of ways, both in training and evaluating with `TxtRayAlign`. With respect to the former, the training curves indicate that a more extensive search of optimal hyperparameters could benefit training behaviour, particularly when training on the full dataset. Furthermore, none of the image encoders used in any of the encoder pairings had been pre-trained on medical images. Using a pair of medically specialised encoders is likely going to make training easier, as domain transfer can be avoided.

With regards to evaluation, the current retrieval is based on a fixed number of sentences. This is unlikely to be suitable in practice, as in reality reports vary in length – a study which reports no findings tends to be shorter. Related to this, sentences are retrieved purely based on a ranking of similarity and does not take into account the magnitude of similarity or similarities between sentences. This can result in irrelevant sentences being retrieved if they are not very "similar" to the image but happen to be the next most similar as well as duplicate sentences being retrieved. Endo et al. [12] propose and discuss a variable-length retrieval mechanism that could provide a step in the right direction. Further, the sentences in the corpus are obtained by a simple split of the reports on any full stops. This results in sentences that can express multiple findings which may limit the specificity of the text the model can retrieve.

Lastly, while we have focused on image-to-text retrieval, `TxtRayAlign` is symmetric between the data modalities and could in principle also perform text-to-image or image-to-image retrieval (see Appendix A), which could improve the searchability of image databases.

9 Conclusion

In this report, we presented exploratory work for generating radiology reports from chest x-rays. `TxtRayAlign` exploits contrastive training to learn similarities between text and images, allowing a retrieval-based mechanism to find reports that are "similar" to an image. The results of our investigation indicate that this approach can help generate reasonably grammatical and clinically meaningful sentences, yet falls short in achieving this with sufficient accuracy. While improvements to the model could be made, our findings are corroborated by others in literature. Besides improving performance, future work could develop other applications of `TxtRayAlign` for other downstream tasks, such as image-to-image or text-to-image retrieval.

References

[1] Eric Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature Medicine*, 25(1):44–56, 2019. ISSN 1546170X. doi:10.1038/s41591-018-0300-7. URL `http://dx.doi.org/10.1038/s41591-018-0300-7`.

[2] Alexander Ke, William Ellsworth, Oishi Banerjee, Andrew Y. Ng, and Pranav Rajpurkar. *CheXtransfer: Performance and parameter efficiency of ImageNet models for chest X-Ray interpretation*, volume 1. Association for Computing Machinery, 2021. ISBN 9781450383592. doi:10.1145/3450439.3451867.

[3] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in Neural Information Processing Systems*, 32(NeurIPS), 2019. ISSN 10495258.

[4] William Boag, Gabriela Berner, and Emily Alesentzer. Baselines for Chest X-Ray Report Generation. pages 1–15, 2019.

[5] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and Curtis P. Langlotz. Contrastive Learning of Medical Visual Representations from Paired Images and Text. pages 1–15, 2020. URL `http://arxiv.org/abs/2010.00747`.

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. 2021. URL `http://arxiv.org/abs/2103.00020`.

[7] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:3156–3164, 2015. ISSN 10636919. doi:10.1109/CVPR.2015.7298935.

[8] Yasuhide Miura, Yuhao Zhang, Emily Tsai, Curtis Langlotz, and Dan Jurafsky. Improving Factual Completeness and Consistency of Image-to-Text Radiology Report Generation. pages 5288–5304, 2021. doi:10.18653/v1/2021.naacl-main.416.

[9] Zhihong Chen, Yan Song, Tsung Hui Chang, and Xiang Wan. Generating radiology reports via memory-driven transformer. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1439–1449, 2020. doi:10.18653/v1/2020.emnlp-main.112.

[10] Jianbo Yuan, Haofu Liao, Rui Luo, and Jiebo Luo. Automatic Radiology Report Generation Based on Multi-view Image Fusion and Medical Concept Enrichment. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11769 LNCS:721–729, 2019. ISSN 16113349. doi:10.1007/978-3-030-32226-7_80.

[11] Baoyu Jing, Pengtao Xie, and Eric P. Xing. On the automatic generation of medical imaging reports. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:2577–2586, 2018. doi:10.18653/v1/p18-1240.

[12] Mark Endo, Viswesh Krishna, and Andrew Y Ng. Retrieval-Based Chest X-Ray Report Generation Using a Pre-trained Contrastive Language-Image Model. *Proceedings of Machine Learning for Health*, pages 209–219, 2021.

[13] Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. 2021. URL `http://arxiv.org/abs/2109.01903`.

[14] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. LiT: Zero-Shot Transfer with Locked-image Text Tuning. 2021. URL http://arxiv.org/abs/2111.07991.

[15] Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques. 2019. ISSN 2331-8422. URL http://arxiv.org/abs/1909.03012.

[16] Hyebin Lee, Seong Tae Kim, and Yong Man Ro. Generation of multimodal justification using visual word constraint model for explainable computer-aided diagnosis. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11797 LNCS:21–29, 2019. ISSN 16113349. doi:10.1007/978-3-030-33850-3_3.

[17] Wilson Silva, Alexander Poellinger, Jaime S Cardoso, and Mauricio Reyes. Interpretability-Guided Content-Based Medical Image Retrieval. In Anne L Martel, Purang Abolmaesumi, Danail Stoyanov, Diana Mateus, Maria A Zuluaga, S Kevin Zhou, Daniel Racoceanu, and Leo Joskowicz, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, pages 305–314, Cham, 2020. Springer International Publishing. ISBN 978-3-030-59710-8.

[18] Zhihong Lin, Donghao Zhang, Qingyi Tac, Danli Shi, Gholamreza Haffari, Qi Wu, Mingguang He, and Zongyuan Ge. Medical Visual Question Answering: A Survey. 2021. URL http://arxiv.org/abs/2111.10056.

[19] Olga Kovaleva, Chaitanya Shivade, Satyananda Kashyap, Karina Kanjaria, Joy Wu, Deddeh Ballah, Adam Coy, Alexandros Karargyris, Yufan Guo, David Beymer Beymer, Anna Rumshisky, and Vandana Mukherjee Mukherjee. Towards Visual Dialog for Radiology. pages 60–69, 2020. doi:10.18653/v1/2020.bionlp-1.6.

[20] Alistair E.W. Johnson, Tom J. Pollard, Seth J. Berkowitz, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih ying Deng, Roger G. Mark, and Steven Horng. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific Data*, 6(1):1–8, 2019. ISSN 20524463. doi:10.1038/s41597-019-0322-0. URL http://dx.doi.org/10.1038/s41597-019-0322-0.

[21] Alistair E. W. 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. MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs. 14:1–7, 2019. URL http://arxiv.org/abs/1901.07042.

[22] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 590–597, 2019. ISSN 2159-5399. doi:10.1609/aaai.v33i01.3301590.

[23] Ruizhe Cheng, Bichen Wu, Peizhao Zhang, Peter Vajda, and Joseph E. Gonzalez. Data-efficient language-supervised zero-shot learning with self-distillation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 3113–3118, 2021. ISSN 21607516. doi:10.1109/CVPRW53098.2021.00348.

[24] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in Neural Information Processing Systems*, 2017-Decem: 1196–1205, 2017. ISSN 10495258.

[25] Akshay Smit, Saahil Jain, Pranav Rajpurkar, Anuj Pareek, Andrew Y. Ng, and Matthew P. Lungren. CheXbert: Combining automatic labelers and expert annotations for accurate radiology report labeling using BERT. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1500–1519, 2020. doi:10.18653/v1/2020.emnlp-main.117.

[26] Mats L. Richter, Wolf Byttner, Ulf Krumnack, Ludwdig Schallner, and Justin Shenk. Size matters. *arXiv*, 2021. ISSN 00114189. doi:10.1177/1043463106060153.

[27] Cade Gordon. train-CLIP, 2021. URL https://github.com/Zasder3/train-CLIP.

A Additional results

A.1 Training loss

Figure 16 shows representative training loss and learning rate curves for the encoder pairings trained on 5% of the data. The training losses first decrease and then increase while the learning rate is relatively high. The loss curves then begin to plateau as the learning rate starts to taper off. This evolution in the loss curves is likely due to addition of the second loss term in Equation 5 which increases when the predicted match probability distributions between the teacher and student differ. Since the teacher is a slowly decaying exponential moving average, the two are more likely to differ when the learning rate is high and the student takes relatively greater update steps.

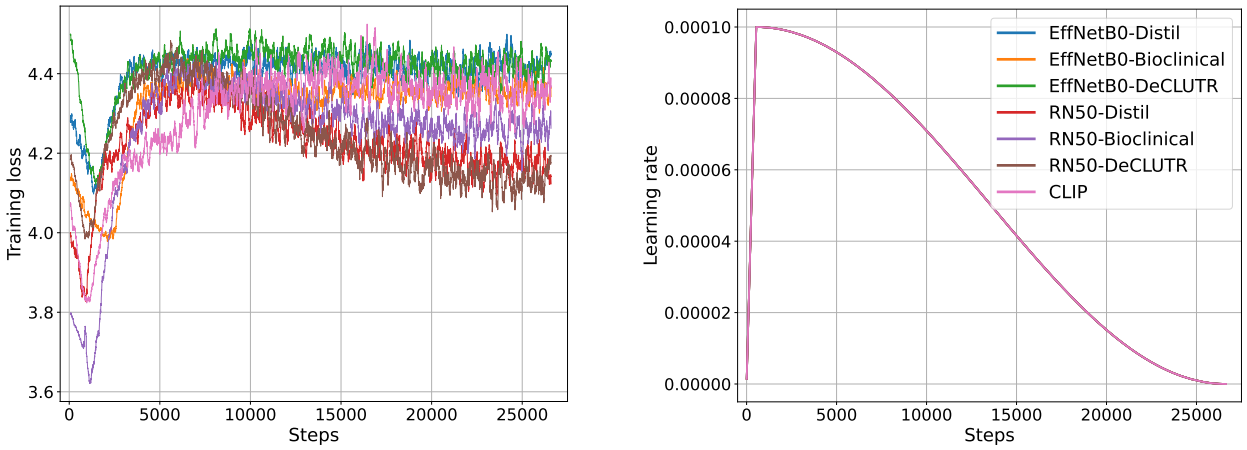


Figure 16: Training loss and learning rate loss on 5% of the data with a batch size of 32.

A.2 Image-to-image retrieval

To illustrate that other downstream tasks are possible, we show results from performing image-to-image retrieval with `TxtRayAlign` in Table 7.

Table 7: Image-to-image retrieval evaluation across 50 test instances. Models trained on 5% of the data.

Encoder						
Image	Text	Duplicates	Flat-hit@2	Precis.@2	Recall@2	F_1@2
EfficientNetB0	DistilBERT	3%	74%	23.6%	41.8%	30.2%
EfficientNetB0	BioclinicalBERT	6%	64%	19.6%	39.3%	26.2%
EfficientNetB0	DeCLUTR	5%	62%	18.4%	40.2%	25.2%
ResNet50	DistilBERT	9%	74%	23.8%	45.4%	31.3%
ResNet50	**BioclinicalBERT**	**5%**	**72%**	**25.1%**	**45.7%**	**32.4%**
ResNet50	DeCLUTR	2%	76%	23.3%	45.2%	30.8%
CLIP-ResNet50	CLIP-Transformer	5%	64%	19.1%	40.1%	25.9%