Radiology Image Captioning

Muthukuda Walawwe Tharindu (2303254), Shanaka Badde Liyanage (2303251), Sabin Bhatta
Project on GitHub:
https://github.com/tharindu326/image-captioning

***Abstract*—This study showcases the application of established methodologies within the Radiology Image Captioning model, an approach that utilizes single encoder and dual encoders models to automate the captioning of radiology images. By leveraging the comprehensive Radiology Objects in Context (ROCO) dataset, which contains over 65,000 images, our model is designed to automatically generate detailed captions that convey the complex features evident in medical imaging. The model integrates conventional Convolution Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks for the encoding of visual elements, while using techniques of feature extraction and Recurrent Neural Networks (RNNs) for caption generation. By adapting these established neural network architectures, our project demonstrates capability in interpreting radiological imagery and producing corresponding textual descriptions. Also the study explores a captioning methodology that includes a retrieval-based model alongside the generative model. This hybrid approach utilizes the training dataset to retrieve a pool of candidate captions. It then evaluates these against the generative model's output, selecting the final caption for an input image based on the highest combined scores from both the generative and retrieval methods.**

Keywords: Radiology Image Captioning, ROCO dataset, Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), Medical Image Analysis, generative-model based image captioning, Retrival-model based image captioning.

I. Introduction

Medical imaging stands as a cornerstone in modern healthcare, providing essential information for the diagnosis and management of numerous health conditions. As the volume of medical images grows, the need for efficient and accurate interpretation becomes increasingly critical. This project introduces an approach to image captioning, a method designed to augment the analysis of these images with descriptive text automatically. In general, image captioning relies on three important steps that apply to all approaches as motioned in [13]. Figure 1

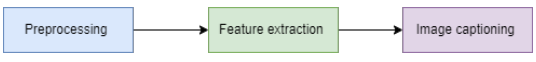

Fig. 1: General steps in image captioning.

Pre-processing: This step involves preparing the unprocessed image for subsequent analysis. Image processing techniques are applied to improve image clarity, thereby accentuating key elements that are critical for generating captions. Pre-processing is crucial as it significantly influences the final caption output, a fact that holds particular importance for medical imagery. At this stage, image augmentation methods might also be employed to expand the dataset size.

Image Feature Extraction: This phase is dedicated to pinpointing and drawing out important and unique characteristics present in the image. The extraction can be accomplished through classical machine learning techniques that focus on specific features or by utilizing deep learning models that autonomously identify features. This may be complemented with methods to reduce the dimensionality of features or with encoding techniques to distill the features that will be used for image description. In cases of limited data, transfer learning is often utilized to adapt features from extensive pre-trained models on large datasets to the specific requirements of the task at hand.

Caption Generation: This stage is about converting the identified features into coherent natural language sentences, considering the syntactic and semantic rules that connect the features. This can be approached by either fetching suitable captions from a database of known captions linked to similar images or by employing predefined rules and templates to construct captions based on the image's features.

Image captioning is a field at the intersection of computer vision and natural language processing that aims to automatically generate descriptive text for images. This capability has transformative potential, particularly for accessibility in technology, content creation, and the interpretation of visual data. The techniques to achieve image captioning are broadly divided into a couple of categories: template based models, generative methods, retrieval methods and hybrid methods.

Generative methods involve creating new captions from scratch. They typically use a combination of Convolutional Neural Networks (CNNs) to interpret the image and Recurrent Neural Networks (RNNs) or Long Short-Term Memory networks (LSTMs) to generate the text. These methods work by training on large datasets of images and their corresponding captions, learning to predict word sequences that are semantically related to the visual content presented. Retrieval methods, by contrast, involve selecting the most appropriate caption from a pre-existing database of captions. Here, the task is to find the best match between the image features and those of the images associated with the pre-written captions. These methods rely on the quality and diversity of the caption database and the effectiveness of the feature matching algorithm.

Template-based image captioning models rely on predefined templates with placeholders filled by detected visual elements from images. Such models are simple and grammatically consistent but lack creativity and depth, often producing generic captions that miss nuanced details.

Hybrid models, in contrast, combine deep learning with

elements of template and retrieval-based methods to generate captions. They use deep learning to analyze image content and construct sentences, sometimes incorporating templates for grammatical structure and retrieval systems for added context. This approach yields more detailed and varied captions, capturing the complexity of images more effectively than template-based systems.

II. Related works

There is a rich body of research that explores various aspects and approaches within the retrieval and generative methodologies. For example, a seminal paper by Vinyals et al., "Show and Tell: A Neural Image Caption Generator,"[15] introduced an influential generative model that uses a deep neural network with a CNN encoder paired with an LSTM decoder. On the retrieval side, research such as "Image Captioning with Semantic Attention" by You et al [18]. demonstrates how attention mechanisms can improve the selection of relevant pre-existing captions by focusing on salient features within the image. Regarding retrieval-based models, Hodosh et al. (2013) [5]: use of retrieval-based models for image captioning, proposing a model that retrieves captions based on the similarity of image features extracted by a deep convolutional neural network (CNN) with those of a large captioned image database. Socher et al. (2014) [14]: developed a dependency tree recursive neural network (DT-RNN) for retrieving image captions, which considers the syntactic structure of sentences to improve the relevance of the retrieved captions. Kiros et al. (2014) [7]: Introduced a combined representation of visual and textual features through a multimodal log-bilinear neural language model, which significantly improved the accuracy of retrieval-based image captioning. (LSTM) network generates the caption. Regarding generative models, enhanced generative models by introducing attention mechanisms that allow the model to focus on specific parts of an image during different stages of caption generation, leading to more detailed and accurate captions was done by Xu et al. (2015) [17]. In Rennie et al. (2017) [12], it proposed a reinforcement learning approach to directly optimize the metric used for evaluation, such as CIDEr, in generative captioning models, which improved the quality of the generated captions. Regarding template-based models, Farhadi et al. (2010) [3] utilized a template-based approach, extracting triplets of objects, actions, and scenes from images and then using these triplets to fill in a template to generate captions. Mitchell et al. (2012) [10] Introduced the idea of using more complex and varied templates that could be filled with detected attributes, objects, and relationships extracted from images. Li et al. (2011) [8] utilized a two-stage template-based model that first creates a coarse draft of the caption using a template and then refines it by substituting more precise attributes and actions detected in the image. Regarding hybrid models, Lu et al. (2018) [9] proposed a neural baby talk model that generates template-like sentences with slots to be filled by detected objects, effectively combining elements of both generative and template-based models. Wang et al. (2019) [16] developed a model that uses a retrieval-based mechanism to select a template and then applies a generative model to refine the caption, thereby incorporating the strengths of both retrieval and generative models.

III. Datasets

The MIMIC-CXR dataset, established by Johnson et al. (2019) [6], is a notable open-source collection featuring 371,920 chest X-rays and associated radiology reports, annotated using the CheXpert labeler and sourced from the Beth Israel Deaconess Medical Center between 2011 and 2016. PadChest, gathered by Bustos et al. (2020) [1], is accessible to the public, comprising 160,868 chest X-rays and 109,931 reports in Spanish, developed with annotations from the Hospital San Juan in Spain, with a notable fraction manually annotated by experts. BCIDR 5, by Zhang et al. (2017) [19], focuses on pathological bladder cancer, consisting of 1,000 annotated images out of whole-slide images from patients, extensively described by pathologists. PEIR Gross, part of the PEIR Digital Library, offers 7,443 images with captions for educational purposes in pathology, curated from a broader collection of 10,000 images, each with a descriptive sentence. The ImageCLEF datasets, designed for ImageCLEF's concept detection and caption prediction tasks, include ImageCLEF caption 2017 by Eickhoff et al. (2017) [2] with 184,614 images, and its expanded version, ImageCLEF caption 2018 by Garcia Seco De Herrera et al. (2018) [4], with 232,305 image-caption pairs, both sourced from PubMed Central. The ROCO dataset, created by Pelka et al. (2018) [11], features 81k radiology images from various medical imaging modalities, curated from PubMedCentral for multimodal image captioning tasks.

IV. Methodology

A. Dataset Generation

We utilized the ROCO dataset for our image captioning project. During preprocessing, we curated a subset of 2,500 radiological images from the dataset and constructed a corresponding vocabulary. The ROCO dataset features a collection of radiological images, each paired with an informative caption. We randomly selected images and systematically organized their captions and file paths into a JSON structure (script:process_dataset.ipynb), which was then used to streamline the data import during the model training phase. In Figure 2 shown a few example entries from the ROCO dataset. (script:data_visualize.ipynb)

B. Vocabulary Generation

To facilitate the encoder-decoder model's understanding of the captions, we converted them into word embeddings using a numerical format. This required the creation of a word-to-index dictionary, which assigns a unique index to each word in the vocabulary. From the subset of 2,500 images, we compiled a vocabulary consisting of 2,200 distinct words. In comparison, when considering the entire dataset of 60,000 images, the vocabulary expanded to encompass

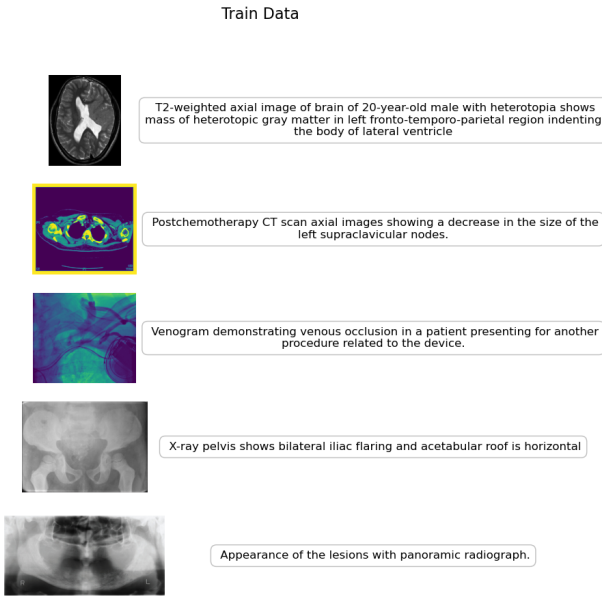

Fig. 2: Visualization of sample images in the dataset.

approximately 14576 unique words. In creating the vocabulary, we incorporated specific tags to signify the beginning, unknown elements, padding, and end of captions. These tags are marked as ¡start¿, ¡unk¿, ¡end¿, and ¡pad¿, respectively. (Script:vocabulary_builder.ipynb)

Additionally, we analyzed the distribution of word usage by calculating the frequency of each word's occurrence. We then charted these frequencies against their corresponding ranks on both logarithmic and linear scales. This allowed us to observe patterns in word behavior and assess their alignment with the power-law distribution. Word occurrences are shown in both linear scale Figure 3 and log scale Figure 4 (Script:vacabulary_frequency.ipynb)

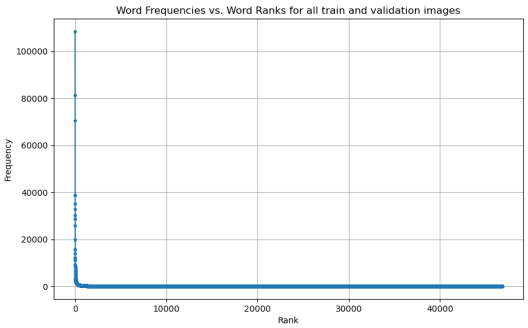

Fig. 3: Word occurrences in linear scale.

C. Word Embeddings

To enhance the analysis of medical terminology, we employed pretrained models specifically designed to understand medical language. Among the available medical language models are BioWordVec, ClinicalBERT, BlueBERT,

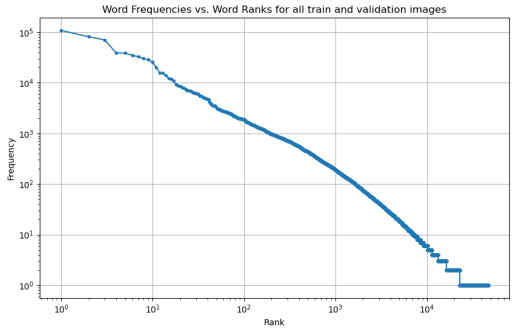

Fig. 4: Word occurrences in log scale.

MedELMo, PubMedBERT, BioBERT, and UMLS Embeddings. We chose the dmis-lab/biobert-base-cased-v1.1 model to generate our word embeddings. To visualize these high-dimensional embeddings, we applied dimensionality reduction techniques such as PCA Figure 5, t-SNE Figure 7, and UMAP Figure 6, ultimately plotting the embeddings in a two-dimensional space (component1 and component2) for better interpretability. (Script:word_embeddings.ipynb)

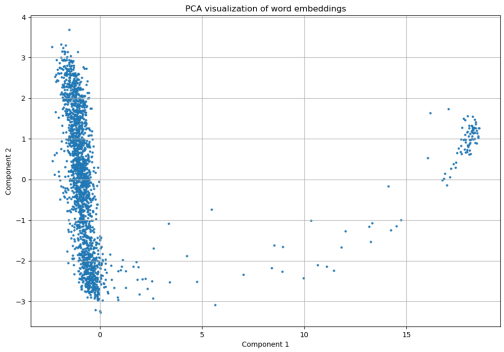

Fig. 5: Word embedding plot using PCA dimensional reduction.

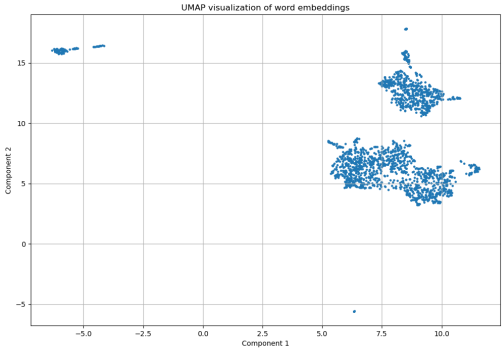

Fig. 6: Word embedding plot using U-MAP dimensional reduction.

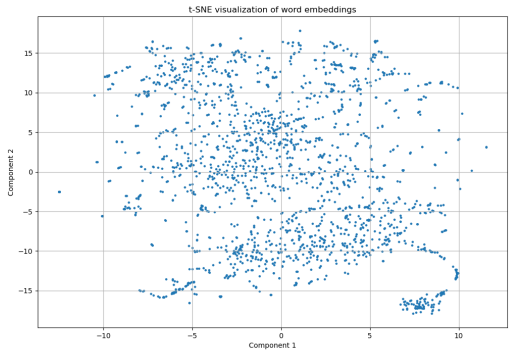

Fig. 7: Word embedding plot using t-SNE dimensional reduction.

D. Data loaders

To begin, we established the ROCODataset class, designed to preprocess the images and captions and to efficiently load them into the GPU/CPU for the encoder's use. Within this class, we first load the images and captions into the data loader. We then tokenize the captions and, by utilizing the constructed vocabulary, transform these captions into numerical sequences. These sequences are subsequently converted into tensors in preparation for input into the model. Due to varying lengths of captions for different images, the batches loaded do not have a uniform size. To standardize the batch tensor dimensions, we employed a collate function that pads the captions, ensuring that each batch conforms to the shape ¡batch size, max caption length¿. In parallel, images are processed and reshaped to meet the requirements of the image encoder, ensuring compatibility and optimal input structure for model training.We utilized PyTorch's built-in DataLoader for implementing the data loading mechanism. (script: data_loader.ipynb)

E. Implementation of generative model

The project is centered around the development of a model that can generate captions for radiology images. The model architecture includes an image encoder based on CNN and a semantic encoder based on LSTM which processes different aspects of the image captions. The image encoder focuses on visual features, while the semantic encoder looks at contextual information. These two streams of information are then integrated by a decoder based on RNN to create meaningful captions. In the field of artificial intelligence, certain algorithms and neural network architectures stand out for their ability to interpret and process complex data. Among these, Long Short-Term Memory networks (LSTMs) and Convolutional Neural Networks (CNNs) are pivotal. LSTMs are a type of Recurrent Neural Network (RNN) that have the capability to learn and remember over long sequences of data. Unlike standard RNNs, LSTMs are designed to overcome the challenge of long-term dependencies by utilizing a series of gates that control the flow of information. These gates allow LSTMs to maintain a memory over time, making them highly effective for tasks involving sequential data such as time series analysis, language processing, and in our case, generating textual descriptions from a series of observed image features. On the other hand, CNNs are predominantly used for analyzing visual imagery. These networks employ a mathematical operation called convolution, which allows them to efficiently process data with a grid-like topology, such as images. CNNs are exceptional at extracting hierarchical features from images, where each layer captures a higher level of abstraction. Due to this property, CNNs have become the backbone of most image recognition and classification tasks. In the context of image captioning, an image encoder typically refers to a CNN that processes an image and converts it into a set of feature vectors that encapsulate the visual details within the image. This encoded representation is then used to generate a caption that describes the content of the image. Conversely, a semantic encoder deals with the processing of textual data, often using models like LSTMs to understand the context and the relationships between words in a sentence. In image captioning, a semantic encoder can be used to incorporate additional context or to process existing captions that can improve the generation of new ones. Finally, an RNN decoder, often employing LSTMs due to their ability to handle sequential data, takes the encoded features from the image and possibly the semantic encoder and generates a sequence of words to form a coherent caption. The RNN decoder does this by predicting each word of the caption in a sequence, sometimes conditioning the prediction of the next word on the previous ones to maintain the context and flow of the sentence. By integrating these components; image encoder, semantic encoders, and RNN decoder, we can construct a model capable of generating accurate and contextually relevant captions for images. Our approach to developing the generative model involved the following methodologies: Initially, we experimented with a single encoder-decoder architecture Figure 8, wherein the image was input into the encoder and the associated captions were input into the decoder, along with the image features extracted by the image encoder. (script:train_single_encoder.ipynb)

F. Single encoder-decoder

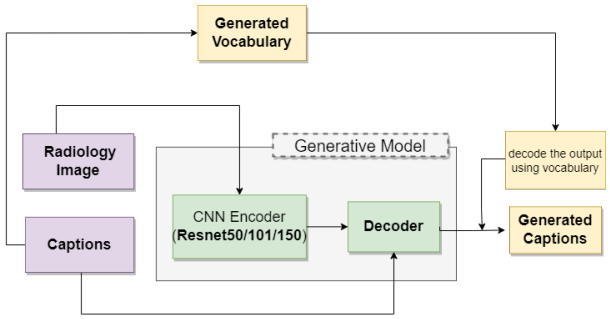

Fig. 8: Single encoder - single decoder generative model architecture.

In a single encoder-decoder architecture for image captioning, the process involves a series of detailed steps: **Preprocessing**: As the first step image resizing is done. The input image is resized to match the input requirements of the Convolutional

Neural Network (CNN). This involves scaling the image to a specific dimension (224x224) while maintaining aspect ratio and possibly applying techniques like cropping or padding to fit the required size. As the second step, caption vectorization is carried out. Accompanying image captions are converted into a vector format. This is achieved by constructing a vocabulary from the training dataset, where each unique word is assigned a specific index. Captions are then tokenized into words, and each word is replaced by its corresponding index in the vocabulary. Special tokens like start-of-sentence (SOS) and end-of-sentence (EOS) are also added to denote the beginning and end of captions. **Encoder**: CNN Feature Extraction is carried out. The resized image is fed into a CNN encoder. The CNN (ResNet-50 backbone) applies a series of convolutional, pooling, and fully connected layers to extract a hierarchy of features from the image. These layers capture various aspects of the image, from basic edges and textures in the early layers to more complex objects and their interactions in the deeper layers. The output of the CNN encoder is a feature map or a flattened feature vector that encapsulates the significant visual aspects of the image, providing a condensed representation for the decoder to process. **Decoder**: The feature representation from the encoder is used as the initial input to the Long Short-Term Memory (LSTM) decoder. The LSTM decoder processes the input in a sequential manner. It starts by receiving the feature vector, which serves as the context for generating the first word of the caption. The LSTM maintains an internal state that is updated as each word is generated, allowing it to keep track of the words it has already produced and the context provided by the image features. At each step, the LSTM outputs a probability distribution over the vocabulary, representing the likelihood of each word being the next word in the caption. **Caption Generation**: The decoder's output at each time step is a vector representing the probability distribution of the next word in the caption. This vector is then used to select the most likely word. The selected word is then converted from its index representation back into its textual form using the reverse of the vocabulary mapping done during preprocessing. This process is repeated for each word until the end-of-sentence token is generated or a maximum sentence length is reached. The sequence of generated words forms the final caption, which ideally describes the content and context of the input image. Encoder decoder model architecture diagrams are shown in Figure 9 and Figure 10. (generated using script:visualize_models.ipynb)

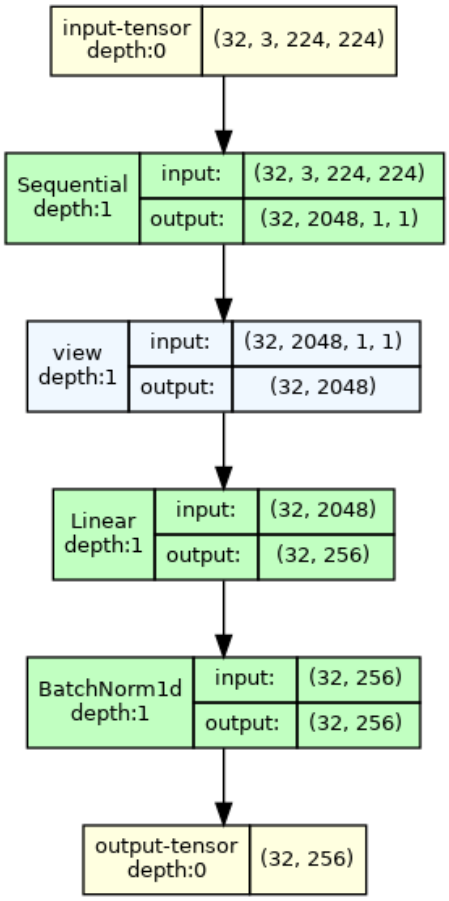

Fig. 9: Encoder model architecture (ResNet-50) in single-encoder-decoder method.

G. Dual encoder-decoder

Subsequently, we evolved our model to incorporate dual encoders Figure 11; one dedicated to processing images and the other to handle semantic encoding of captions. The outputs of these two encoders, namely the image features and the semantic caption features, were merged and then supplied to the decoder.(script:train_duel_encoder.ipynb)

After completing the preprocessing steps similar to those in the single encoder-decoder model, the image and its corresponding caption are independently inputted into two distinct encoders. **Image Encoder**: The resized image is inputted into a CNN (ResNet-50 backbone) encoder to extract a rich set of features. **Semantic Encoder**: Parallel to the image encoding, the vectorized captions are processed through an LSTM-based semantic encoder. This LSTM is designed to capture the semantic and syntactic structure of the text. The LSTM processes the sequence of word vectors, updating its internal state at each step, to produce a semantic feature representation of the caption. This representation captures the essence and context of the text. **Fusion of Features**: The next step involves combining the features from both the image encoder and the semantic encoder. This is done by concatenating the feature vectors from both encoders. The fused feature vector now contains a comprehensive representation that encompasses both visual and textual information, providing a rich context for the decoder. After the encoder feature fusion, the output feature tensor is fed to the decoder like in the single decoder encoder model and then caption generation is done. Semantic encoder, image encoder and decoder model architectures are defined in Figure 12 and Figure 13. (generated using script:visualize_models.ipynb)

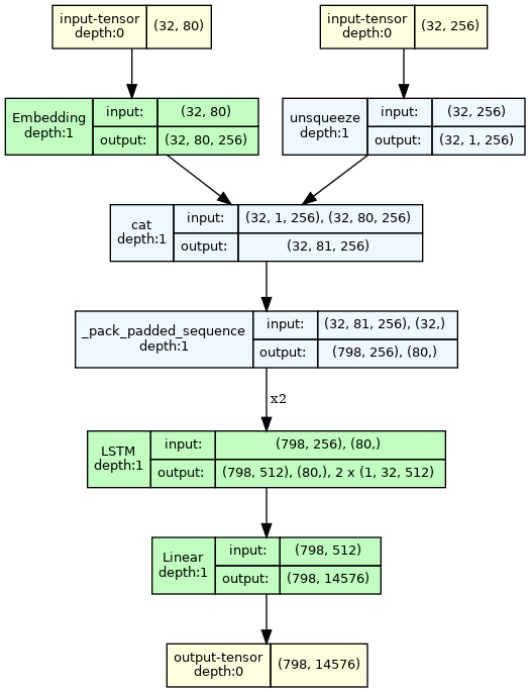

Fig. 10: Decoder model architecture in single-encoder-decoder method.

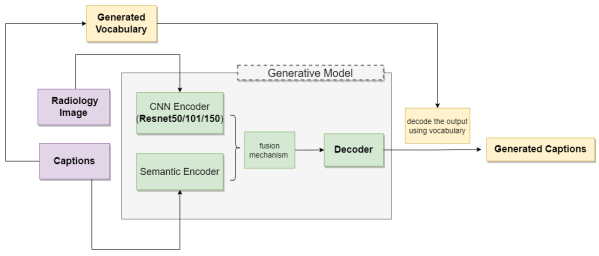

Fig. 11: Generative model architecture of image encoder, semantic encoder and decoder.

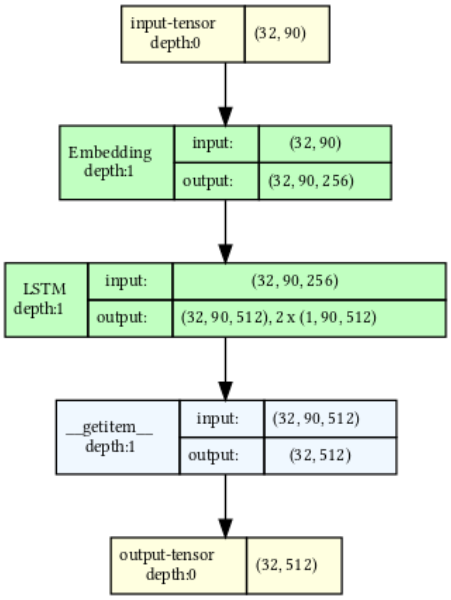

Fig. 12: Semantic Encoder model architecture in dual-encoder-decoder method.

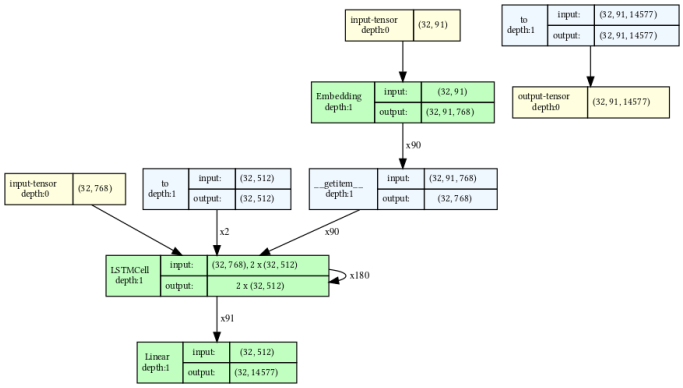

Fig. 13: Decoder model architecture in dual-encoder-decoder method.

H. Implementation of retriever model

To enhance the accuracy of the captions, the project employs a dual-method approach. Initially, captions are generated using the encoder-decoder model. Following this, a retrieval method is also applied Figure 14, which selects the most appropriate caption from a set of pre-existing captions. The final stage involves comparing the generated and retrieved captions to determine which one closely aligns with the ground truth captions defined as the most accurate caption for the image. (Script:similarity_single_encoder.ipynb)

I. Similarity between captions

In our retrieval-based approach, we aimed to gauge the similarity between the generated caption (Y) and the reference or ground truth caption (X). This process also assists in fetching the most closely aligned caption for the generated caption (Y) from our training dataset.(Z) To assess caption similarity, we initially employed embedding techniques to derive word embeddings through models like Word2Vec, fastText, and BioBERT. Subsequently, we utilized various similarity metrics such as cosine similarity, Jaccard similarity, and the BLEU score to calculate the proximity or dissimilarity between these word embeddings. Furthermore, we harnessed a CountVectorizer to convert the captions into vector forms, upon which we applied cosine similarity measures to identify the most accurate matching caption within the dataset. After obtaining the Y (generated caption) and Z (a set of potential true captions), we compared them with ground truths to identify the most accurate match for the image, which we determined by the highest similarity score. (Script: similarity_single_encoder.ipynb)

V. Results

The single encoder model underwent training on 60,000 images for 20 epochs, featuring a hidden size of 512, which refers to the dimensionality of the internal state vectors, and an embed size of 256, which is the size of the vectors used

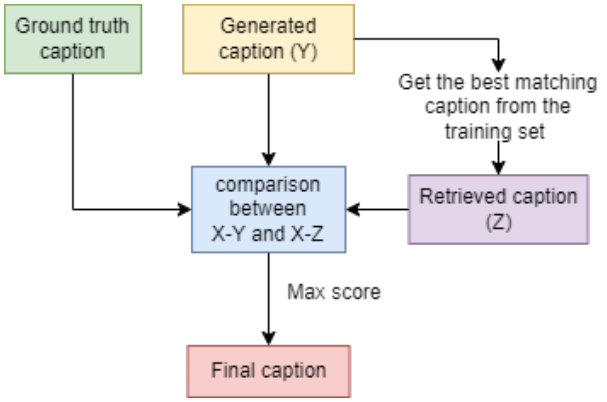

Fig. 14: Model fusion of generative model and retrieved model.

to represent words. (Script: train_single_encoder.ipynb) The model demonstrated improved accuracy, with the generative model's predicted captions closely aligning with the corresponding images. Two examples are illustrated in the figures Figure 15 Figure 16.

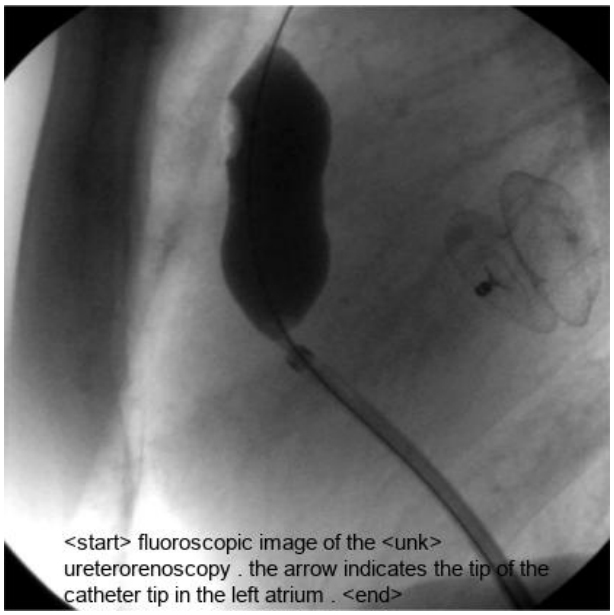

Fig. 15: sample output of single encoder model.

Regarding the retrieval model, the BLEU score was utilized to identify the best matches from the training set. The caption with the highest match score to the generated caption (Y) was chosen as the retrieval caption (Z). An example of this outcome is displayed in the figure. In the model fusion phase, the maximum score from the comparison of the ground truth caption (X) with both the generated caption (Y) and the retrieved caption (Z) determined the final caption for the image (Script:similarity_single_encoder.ipynb). The process of selecting the final caption for the image is illustrated in the figure Figure 17 .

The dual-encoder model underwent training on a curated set of 2,500 images.(Script: train_duel_encoder.ipynb) It was

Fig. 16: sample output of single encoder model.

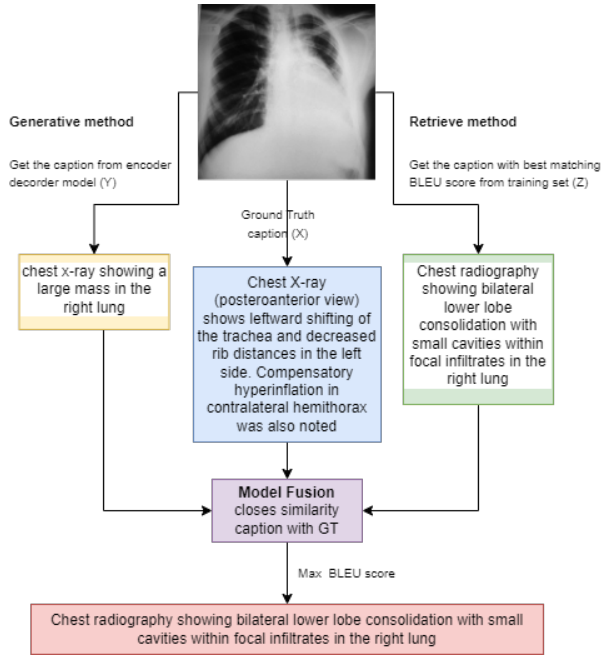

Fig. 17: sample output for final caption generation.

configured with a hidden size of 512 and an embedding size of 256. Below are some of the outcomes of this training. (Script: similarity_duel_encoder.ipynb) The results (Figure 18, Figure 19) are not accurate compared to the single encoder method.

Fig. 18: sample output of dual encoder model.

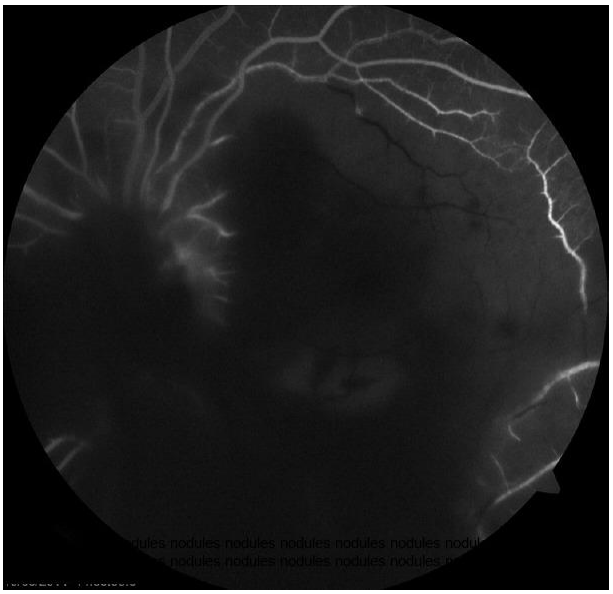

Fig. 19: sample output of dual encoder model.

VI. Evaluation

We monitored the training and validation losses to gauge the performance of both dual and single encoder models. Validation loss measures the model's performance on a set of data not seen during training, reflecting its ability to generalize. Training loss, on the other hand, indicates how well the model is learning from the data it has been trained on. For the single encoder model, both training and validation losses decreased steadily, suggesting consistent learning across the training set and validation set. However, in the dual encoder model, while the training loss continued to decline, indicating improved learning from the training data, the validation loss exhibited variability, hinting at potential over-fitting or instability in the model when applied to new data. The figure illustrates (single encoder: Figure 20/Figure 21, dual encoders: Figure 22/Figure 23) the trends in training and validation losses for both models.

Fig. 20: Training loss of single encoder model architecture.

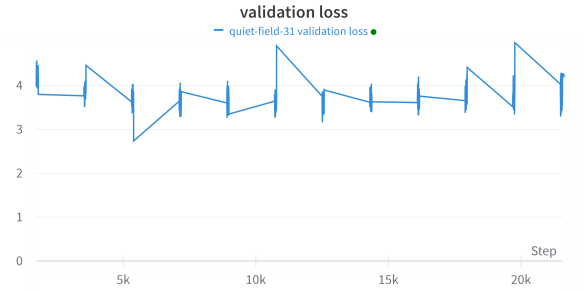

Fig. 21: Validation loss of single encoder model architecture.

Fig. 22: Training loss of dual encoder model architecture.

Test data validation is a critical process of evaluating a machine learning model's performance on an independent dataset that was not used during training. It's essential for assessing the model's ability to generalize to new, unseen data. The importance of test data validation lies in its ability to provide an unbiased assessment of a model's predictive power. It helps identify issues like over-fitting, where a model might perform well on the training data but poorly on new data, thus ensuring the robustness and reliability of the model in practical applications. In the context of image captioning, there are various metrics used to evaluate the accuracy of

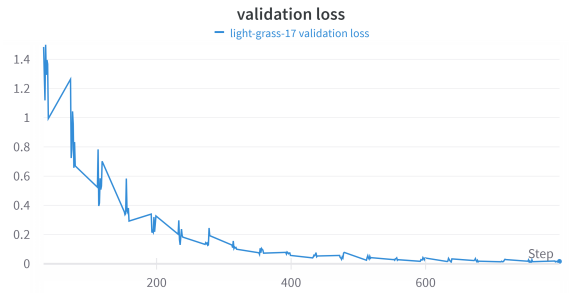

Fig. 23: Validation loss of dual encoder model architecture.

the models by comparing their predicted captions against the ground truth. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measures the overlap of n-grams between the predicted and ground truth captions, focusing on recall. BLEU (Bilingual Evaluation Understudy) metric evaluates the quality of text translation based on n-gram precision, with penalties for too-short sentences. METEOR (Metric for Evaluation of Translation with Explicit Ordering) aligns the predicted caption with the reference, accounting for synonymy and stemming, and then calculates precision and recall. SPICE (Semantic Propositional Image Caption Evaluation) metric assesses the semantic fidelity of a caption by comparing the scene graph of the predicted caption with that of the reference. BERTScore leverages the pre-trained contextual embeddings from BERT and matches words in predicted and reference captions for semantic similarity scoring. BLEURT metric uses BERT embeddings as well, but it is trained on human ratings to predict the quality of text generation more accurately. The table summarizes the performance of the test dataset across all these metrics for both dual and single encoder models, providing insights into how each model fares in terms of accuracy, semantic relevance, and linguistic quality of the generated captions. Evaluation matrices for both methods have shown in Table I

TABLE I: Evaluation metrics for test data evaluation

matric	dual encoder model	single encoder model
rouge-1	0	0.181
rouge-2	0	0.049
rouge-l	0	0.162
BLEU	0	0.019
METEOR	0	0.131
SPICE	0.002	0.063
BERT-precison	0.742	0.862
BERT-recall	0.78	0.842
BERT-f1	0.761	0.851
BLEURT	0.157	0.243

VII. Discussion

The Table I compares the performance of a dual-encoder model and a single encoder model on various metrics used. ROUGE metrics evaluate the overlap of n-grams between the generated text and a reference text. The single encoder model outperforms the dual-encoder model significantly in all three metrics (ROUGE-1: 0.181 vs. 0, ROUGE-2: 0.049 vs. 0, ROUGE-L: 0.162 vs. 0). This indicates better overlap of unigrams, bigrams, and longest common subsequences, respectively, for the single encoder model. metric evaluates the correspondence of n-grams between the generated and reference texts, with a focus on precision. The single encoder model scores 0.019, while the dual-encoder model scores 0. This suggests that the single encoder model is better at producing n-grams that are present in the reference texts. BERT metrics ((Precision, Recall, F1)) are based on BERT embeddings to evaluate semantic similarity. Even though we consider the semantic similarity in duel encoder, it performs reasonably well (Precision: 0.742, Recall: 0.78, F1: 0.761) but the single encoder model still outperforms it (Precision: 0.862, Recall: 0.842, F1: 0.851). Overall, these results indicate that the single encoder model consistently outperforms the dual-encoder model across a range of metrics, suggesting it is more effective in generating text that aligns closely with reference texts in terms of both lexical and semantic content.

Various factors might contribute to a dual encoder-decoder model being less accurate compared to a single encoder-decoder model. Dual encoder-decoder models involve integrating two separate encoders, which can be more complex than using a single encoder. This complexity can lead to difficulties in effectively combining the features extracted by each encoder, potentially reducing the overall accuracy. Further the complexity with higher number of training parameters can make them more prone to over-fitting, especially when the training data is not sufficiently diverse or large (2000 images were used). Ensuring that the image encoder and the semantic encoder are well-aligned and effectively synchronized can be challenging. Any misalignment between these encoders can lead to inaccurate or irrelevant captions. In a dual encoder model, balancing the representation of features from the image and the semantic context is critical. If one encoder dominates the other, the generated captions might not accurately reflect the content of the image or the intended semantic nuances. Also training a dual encoder-decoder model involves optimizing two separate encoders along with the decoder. This can complicate the optimization process, making it harder to find a model configuration that produces the most accurate results. On the other hand, dual encoder models might struggle more with data sparsity. The limited size of the dataset, comprising only 2000 images, can also be a significant factor contributing to the lower accuracy of a dual encoder-decoder model compared to a single encoder-decoder model. A smaller dataset may not provide enough variation and examples for the more complex dual encoder-decoder system to effectively learn and generalize from, leading to reduced performance.

Looking ahead to future enhancements in model training, several strategies can be utilized. A pivotal factor in model performance is the dataset's size and diversity. Enlarging the dataset and incorporating a wider range of examples can provide the model with a more comprehensive learning experience, crucial for the complex nature of dual encoders. The effectiveness of pre-trained models can be significantly amplified

by fine-tuning them on data closely aligned with the specific application. This approach ensures that the model is better adapted to the nuances of the task at hand.Investigating various techniques for merging features from both the image and text encoders is essential. Methods like concatenation, element-wise addition, or advanced attention mechanisms might lead to more effective integration of multi-modal data. A common issue in models with numerous parameters is over-fitting. Regularization methods such as dropout, L1/L2 regularization, or early stopping are some options we can include to overcome the over-fitting. Modifying key hyper-parameters, including the learning rate, batch size, and the architecture specifics of the encoder and decoder, is crucial. Automated techniques like grid search or Bayesian optimization can facilitate this process. The use of attention mechanisms can aid the model in focusing on relevant segments of the image and text, thus enhancing the quality of the feature representation. Augmenting the existing dataset through various transformations in images and text can artificially expand the dataset, thereby improving the model's ability to generalize. Additionally, adopting alternative approaches, such as template-based models, could further refine the accuracy. Compared to standalone generative or retrieval methods, the combined approach typically yields better results.

There are several challenges and limitations associated with the implementation. One of the primary challenges is capturing the context and deeper meaning of an image. While models can identify objects and actions, understanding the context, relationships, and subtleties (like anatomy, radiology aspects, conditions) is difficult. The accuracy and effectiveness of an image captioning model heavily depend on the quality and diversity of its training data. Datasets may not cover all possible scenarios, leading to biases or limited understanding of less-represented subjects or environments. Also Generating captions that are not only accurate but also semantically rich and detailed is challenging. Often, captions might be too generic or miss nuanced details that a human observer might easily recognize.

VIII. Conclusion

The objective of our project was to create a model capable of generating relevant captions for radiology images. Leveraging the ROCO dataset, we implemented an architecture that utilized single and dual encoders for image and semantic features. The project's findings reveal that the marriage of domain-specific embeddings with a generative model holds substantial promise for the task of medical image captioning. We explored both single and dual encoder models for the task of image captioning and incorporated a retrieval method to enhance the caption generation process. In the end, we merged the outputs from both models to derive the final captions. Despite the sophisticated architecture of the dual encoder model, its performance was limited by the small size of the image dataset. Conversely, the single encoder model demonstrated relatively higher accuracy. To assess the efficacy of these models, we employed various evaluation metrics such as ROUGE, BLEU, METEOR, SPICE, BERT, and BLEURT. These measures provided a comprehensive comparison of the effectiveness of each method, allowing us to determine the most accurate approach for image captioning in our study. There are various methods to enhance the accuracy of image captioning, and among them, hybrid methods often yield higher accuracy because they integrate the strengths of both retrieval-based and generative models. This combination allows for the flexibility and creativity of generative approaches, as well as the precision and relevance that retrieval-based systems offer by pre-existing captions. Consequently, hybrid methods benefit from the robustness and diversity of combined techniques to produce more detailed descriptions for images.

A. Additional script files

All the Scripts related to tasks are listed in Table II with their description.

TABLE II: scripts

Script	Description
process_dataset.ipynb	Task1: select dataset and generate the json data object
data_visualize.ipynb	Task1: visualize samples from the dataset.
vocabulary_builder.ipynb	Task2: build the vocabulary.
vacabulary_frequency.ipynb	Task3: plot the word occurrences.
word_embeddings.ipynb	Task4: generate word embeddings (using different methods) and plotting.
data_loader.ipynb	Task5: pytorch data loading functions generative model (encoder-decoder models)
train_duel_encoder.ipynb	Task5: fit data: model training of duel encoder model
train_single_encoder.ipynb	Task5: fit data: model training of single encoder model
similarity_single_encoder.ipynb	Task6 - Task9 using single encoder generative model Task6: get generative captions and similarities with GTs (X) using different similarity matrices Task7: Retrieval method: get the most similar caption for generated caption (Y) from training set (Z) Task8: model fusion: compare GT (X) ' with (Y) and (Z) and get the best caption as (Y) or (Z) and assigned it to test image Task9: Evaluation metrics
similarity_duel_encoder.ipynb	Task6 - Task9 using dual encoder generative model
inference.ipynb	Task6 - inference of dual encoder model
visualize_models.ipynb	visualization of model architectures

References

[1] Aurelia Bustos et al. "PadChest: A large chest x-ray image dataset with multi-label annotated reports". In: *Medical Image Analysis* 66 (Aug. 2020), p. 101797. DOI: 10.1016/j.media.2020.101797.

[2] Carsten Eickhoff et al. "Overview of ImageCLEFcaption 2017 – Image Caption Prediction and Concept Detection for Biomedical Images". In: Jan. 2017.

[3] Ali Farhadi et al. "Every Picture Tells a Story: Generating Sentences from Images". In: vol. 6314. Sept. 2010, pp. 15–29. ISBN: 978-3-642-15560-4. DOI: 10.1007/978-3-642-15561-1_2.

[4] Alba García Seco de Herrera et al. "Overview of the ImageCLEF 2018 caption prediction tasks". In: Sept. 2018.

[5] Micah Hodosh, Peter Young, and Julia Hockenmaier. "Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics". In: *Journal of Artificial Intelligence Research* 47 (May 2013), pp. 853–899. DOI: 10.1613/jair.3994.

[6] Alistair E. W. Johnson et al. *MIMIC-CXR-JPG, a large publicly available database of labeled chest radiographs*. 2019. arXiv: 1901.07042 [cs.CV].

[7] Ryan Kiros, Ruslan Salakhutdinov, and Richard Zemel. "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models". In: *31st International Conference on Machine Learning, ICML 2014* 3 (Nov. 2014).

[8] Y Li et al. "Substitution of wheat dried distillers grains with solubles for barley grain or barley silage in feedlot cattle diets: Intake, digestibility, and ruminal fermentation". In: *Journal of animal science* 89 (Mar. 2011), pp. 2491–501. DOI: 10.2527/jas.2010-3418.

[9] Jiasen Lu et al. *Neural Baby Talk*. 2018. arXiv: 1803.09845 [cs.CV].

[10] Cameron Mitchell et al. "Resistance exercise load does not determine training-mediated hypertrophic gains in young men". In: *Journal of applied physiology (Bethesda, Md. : 1985)* 113 (Apr. 2012), pp. 71–7. DOI: 10.1152/japplphysiol.00307.2012.

[11] Obioma Pelka et al. "Radiology Objects in COntext (ROCO): A Multimodal Image Dataset: 7th Joint International Workshop, CVII-STENT 2018 and Third International Workshop, LABELS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings". In: Sept. 2018, pp. 180–189. ISBN: 978-3-030-01363-9. DOI: 10.1007/978-3-030-01364-6_20.

[12] Steven J. Rennie et al. *Self-critical Sequence Training for Image Captioning*. 2017. arXiv: 1612.00563 [cs.LG].

[13] Beddiar Romaissa, Mourad Oussalah, and Tapio Seppänen. "Automatic captioning for medical imaging (MIC): a rapid review of literature". In: *Artificial Intelligence Review* 56 (Sept. 2022). DOI: 10.1007/s10462-022-10270-w.

[14] Richard Socher et al. "Grounded Compositional Semantics for Finding and Describing Images with Sentences". In: *Transactions of the Association for Computational Linguistics* 2 (Dec. 2014), pp. 207–218. DOI: 10.1162/tacl_a_00177.

[15] Oriol Vinyals et al. *Show and Tell: A Neural Image Caption Generator*. 2015. arXiv: 1411.4555 [cs.CV].

[16] Xin Wang et al. *VATEX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research*. 2020. arXiv: 1904.03493 [cs.CV].

[17] Jungang Xu, Hui Li, and Shilong Zhou. "An Overview of Deep Generative Models". In: *IETE Technical Review* 32 (Dec. 2014), pp. 131–139. DOI: 10.1080/02564602.2014.987328.

[18] Quanzeng You et al. *Image Captioning with Semantic Attention*. 2016. arXiv: 1603.03925 [cs.CV].

[19] Zizhao Zhang et al. "TandemNet: Distilling Knowledge from Medical Images Using Diagnostic Reports as Optional Semantic References". In: Sept. 2017, pp. 320–328. ISBN: 978-3-319-66178-0. DOI: 10.1007/978-3-319-66179-7_37.