FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Automatic Generation of Medical Reports

Patrícia Ferreira Rocha

Mestrado Integrado em Bioengenharia

Supervisor: Luís Filipe Pinto de Almeida Teixeira

Second Supervisor: Isabel Cristina Rio-Torto de Oliveira

July 28, 2020

Automatic Generation of Medical Reports

Patrícia Ferreira Rocha

Mestrado Integrado em Bioengenharia

July 28, 2020

Resumo

As tecnologias de Deep Learning (DL) têm sido amplamente implementadas nas mais variadas tarefas devido à sua capacidade em processar grandes volumes de dados. As Redes Neuronais Convolucionais (CNNs), em particular, destacam-se em tarefas visuais, como o diagnóstico através de imagens médicas. Apesar das suas vantagens, os algoritmos de DL são modelos de caixa-preta, ou seja, o seu alto desempenho tem o custo de alta abstração. Embora em algumas circunstâncias os problemas relacionados com a caixa-preta sejam menos relevantes, em domínios de alto risco, como os cuidados de Saúde, o alto desempenho é insuficiente. Um diagnóstico baseado em imagem é geralmente acompanhado de um relatório médico detalhado descrevendo o conteúdo da mesma. Analogamente, um modelo deve ser capaz de apresentar explicações textuais para as suas previsões.

De forma a contornar estas limitações, este trabalho envolveu a validação de modelos previamente estabelecidos de geração de texto a partir de imagens, usando raios X e os relatórios médicos correspondentes. No geral, estes modelos consistem numa arquitetura combinada de uma CNN para codificar o conteúdo da imagem em vetores de características e uma rede neuronal Long Short-Term Memory (LSTM) para descodificar os ditos vetores numa sequência de palavras. Este trabalho pretende ser uma primeira abordagem no sentido de implementar um sistema que produza explicações textuais auxiliares em problemas com dados visuais, como o diagnóstico baseado em imagem, de forma não supervisionada.

Embora os métodos baseados em DL se demonstrem promissores na área, há ainda espaço para melhorias, uma vez que os relatórios médicos gerados devem fornecer evidências e detalhe suficientes para suportar uma decisão médica. Contudo, os métodos de estado-da-arte em geração automática de relatórios médicos são ainda incapazes de diversificar a linguagem e tendem a falhar informação visual vital. Assim, há ainda um longo caminho a percorrer para se conseguir gerar explicações textuais sem supervisão direta.

Abstract

Deep Learning (DL) technologies have been widely adopted due to their ability to handle large amounts of data. Convolutional Neural Networks (CNNs), in particular, excel in visual tasks such as image-based diagnosis. Despite their well-known advantages, DL algorithms are black-box models. In other words, the high performance comes at the price of high abstraction. While in certain fields the black-box-related issues are less relevant, in high-risk domains, such as Healthcare, generating accurate predictions is no longer enough. An image-based diagnosis is usually coupled with a detailed medical report describing the findings. Similarly, a model should be able to present textual explanations for its predictions.

To address these shortcomings, this work involved the validation of previously established models for image captioning on a set of chest X-rays paired with the corresponding medical reports. Overall, these models are based on a combined architecture of a CNN to encode the image content into smaller feature vectors and a Long Short-Term Memory (LSTM)-based Recurrent Neural Network (RNN) to decode such vectors into a sequence of words. This work is intended to be a first approach towards the end goal of implementing a pipeline that produces auxiliary textual explanations for problems using visual data, such as image-based diagnosis, in an unsupervised fashion.

Although DL-based methods are extremely promising in this field, there is still space for improvement since the generated medical reports should be able to provide enough evidence and detail to support a medical decision. Nevertheless, state-of-the-art methods on medical imaging report generation are still incapable of diversifying language and typically miss vital visual information, meaning there is still a long way ahead before being able to generate textual explanations without direct supervision.

Agradecimentos

Aos meus orientadores, Prof. Doutor Luís Teixeira e Isabel Rio-Torto, por toda a orientação e disponibilidade ao longo do desenvolvimento deste trabalho. Foi um prazer poder trabalhar convosco.

Aos meus pais por me terem dado sempre as melhores condições possíveis e pelo voto de confiança. Às minhas tias por estarem sempre à distância de uma chamada e acreditarem mais em mim do que eu própria. Ao meu irmão Pedro — o meu maior crítico — por ser uma boa companhia, apesar de só pensar em futebol e miúdas. Tenho ainda que vos agradecer a todos pela vossa compreensão e paciência nos inúmeros fins-de-semana em que vim a casa, mas não estava realmente cá.

Aos amigos de sempre por continuarem a tolerar o meu mau feitio tantos anos depois. Aos amigos que conheci na faculdade por todos os serões em que por pouco não nos agredimos verbalmente e pela amizade que superou todas as entregas no último minuto.

Finalmente, à FEUP por cinco anos que, embora absolutamente disfuncionais, foram decisivos para o meu crescimento. A todos, o meu maior agradecimento.

Patrícia Rocha

"Só se nos detivermos a pensar nas pequenas coisas
chegaremos a compreender as grandes."

José Saramago

Contents

List of Figures

List of Tables

Abbreviations and Symbols

ANN	Artificial Neural Network
BCIDR	Bladder Cancer Image and Diagnostic Report
BioBERT	Bidirectional Encoder Representations from Transformers for Biomedical Text Mining
BLEU	Bilingual Evaluation Understudy
BPTT	Backpropagation Through Time
CAV	Concepts Activation Vectors
CIDEr	Consensus-based Image Description Evaluation
CNN	Convolutional Neural Network
CV	Computer Vision
DL	Deep Learning
DNN	Deep Neural Network
EU	European Union
GDPR	General Data Protection Regulation
gLSTM	Guided Long Short-Term Memory
GRU	Gated Recurrent Unit
HIPAA	Health Insurance Portability and Accountability Act
IU	Indiana University
K-NN	K-Nearest Neighbours
LIME	Local Interpretable Model-Agnostic Explanations
LRP	Layer-Wise Relevance Propagation
LSTM	Long Short-Term Memory
MAPLE	Model Agnostic Supervised Local Explanations
METEOR	Metric for Evaluation of Translation with Explicit Ordering
ML	Machine Learning
MLC	Multi-label Classification
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NLG	Natural Language Generation
NLP	Natural Language Processing
OpenI	Open Access Biomedical Image Search Engine
PEIR	Pathology Education Informational Resource
PMC	PubMed Central
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
SA	Sensitivity Analysis
SGD	Stochastic Gradient Descent
SHAP	Shapley Additive Explanations
SPICE	Semantic Propositional Image Caption Evaluation
TCAV	Testing with Concept Activation Vectors
TF-IDF	Term Frequency-Inverse Document Frequency
XAI	Explainable Artificial Intelligence

Chapter 1

Introduction

1.1 Context and Motivation

Deep Learning (DL) technologies have become increasingly adopted due to their ability to select information and recognize patterns in an autonomous fashion. Additionally, DL tends to offer better performance than classical Machine Learning (ML) algorithms when dealing with a large amount of data. Therefore, DL is being applied as a solution for a variety of problems, including in the medical domain. Medical imaging and diagnostics are fields in which Convolutional Neural Networks (CNNs), in particular, are of great significance. CNNs extract meaningful information from images to obtain a classification that translates into a diagnosis.

Despite their advantages, DL algorithms are known as black-box models since they are extremely complex and usually trained end-to-end, with minimal human interference. In other words, these models take the raw input and automatically transform it into a feature space, without the need for feature engineering and domain knowledge. Unfortunately, the high accuracy of these algorithms is achieved through this higher abstraction. This limitation has led to increased interest in understanding the reasoning behind the decisions made by these systems, that is, a "*collection of features of the interpretable domain, that have contributed for a given example to produce a decision*" [1]. This concept of explanation was defined by Montavon *et al.*

While in certain application fields the black-box-related issues are less critical, in high-risk domains such as Healthcare, it is crucial to interpret, understand, and explain the model's behavior. Healthcare is an area in which explainable models are a key concept to improve trust and transparency. In most real-world tasks, it is no longer enough to generate accurate predictions but also understand the root causes behind the decision. An image-based diagnosis is usually associated with a medical report describing the findings that led to such a conclusion. The model should be able to present an explanation, textual for instance, that ideally matches the findings in such reports.

There is an upsurge of interest in finding a solution that eliminates the high abstraction-related limitation. However, this is a relatively new research area that faces a few challenges. First, there is

a shortage of large medical datasets with accurate annotations made by experts that allow the network to capture their knowledge — and it would be rather time-consuming to manually build one. Moreover, sharing clinical data has also become increasingly difficult due to the progressively protective patient privacy laws. The second challenge pertains to the quality of an explanation, which is highly subjective since it depends on many factors, such as the domain and/or the end-user. There is still no groundbreaking metric to assess an explanation, making it difficult to evaluate the model and collect data that includes the annotation for what a good explanation should be.

1.2 Problem Description and Objectives

Medical imaging is a fundamental tool used for providing a diagnosis. However, the current methods for assisting physicians are not foolproof. In that sense, some questions arise: do we know how the network learns? Do we know if and when it fails?

Accordingly, this work intends to be a first approach towards implementing a system that produces auxiliary textual explanations for visual problems such as image-based diagnosis. Image captioning techniques and a set of medical images paired with the corresponding text will be used as an approach to improve reliability and the informative quality of medical reports.

A detailed medical report is usually long and comprises multiple sentences, each focusing on a different topic or image region. For this reason, medical report generation is more complex than traditional image captioning. Image captioning techniques are usually designed to generate shorter and less complex text while medical reports consist of longer pieces of text, with a varying number of sentences and words. Additionally, image captioning frequently prioritizes readability over descriptive accuracy. However, reports must be not only readable and grammatically correct, but also clinically accurate.

This work involves the validation of previously established image captioning models on a medical image dataset. Overall, these models are based on an encoder-decoder architecture. The encoder stage should be able to analyze the visual content of the image and extract the features that preserve the most information. The decoder should then take the encoded image and generate the textual explanation. Hence, the system will rely on a CNN and a Recurrent Neural Network (RNN), namely a Long Short-Term Memory (LSTM)-based RNN. Both networks will be further detailed in the following sections.

1.3 Document Structure

This document is organized as follows: Chapter 1 offered an introduction to the field of interest of this work. Chapter 2 provides background information to subsequent chapters. It introduces some relevant DL- and CNN-related concepts (sections 2.1 and 2.1.2, respectively), detailing the most widely adopted CNN architectures. Chapter 3 briefly addresses explainability concepts, their relevance, and application fields. A few methods of explainability in DL are also presented. Chapter 4 contemplates the evolution of natural language generation (NLG) and the methods widely

adopted in tasks involving sequential data (section 4.1). Then, state-of-the-art and literature techniques for image captioning (section 4.2) and medical imaging report generation (section 4.4) are reviewed, as well as the commonly used datasets (section 4.5) and evaluation metrics (section 4.3). Chapter 5 outlines the proposed approach and Chapter 6 reports the corresponding results. Lastly, Chapter 7 draws conclusions on the work done and offers potential future research paths.

Chapter 2

Image Classification in Deep Learning

This Chapter focuses on DL technologies, beginning with an overview of what DL is. Then, some fundamental concepts common to any DL system, such as weight initialization, activation and loss functions, regularization techniques, and transfer learning are introduced. The remainder of the Chapter discusses CNNs, namely the building blocks and the evolution of the most widely adopted architectures, demonstrating the increasing depth of such representations.

2.1 Deep Learning

DL is a branch of ML that is based on Artificial Neural Networks (ANNs). ANNs were inspired by the way the brain learns. In the 1960s, the *perceptron* model was introduced by Frank Rosenblatt [2]. This model consists of only one neuron that takes multiple inputs and computes a single output (figure 2.1). Since then, Rosenblatt's single-layer perceptron model has been succeeded by models with added complexity, the Multilayer Perceptrons (MLPs).

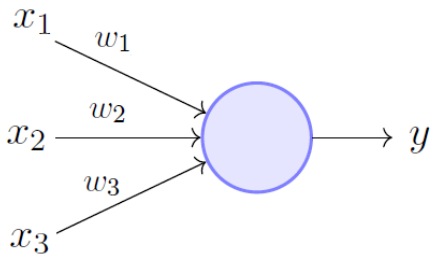


Figure 2.1: Perceptron model. Single-layer model that takes multiple inputs and computes a single output. Extracted from [3].

Unlike traditional ML models, DL is able to perform automatic feature extraction — it takes the raw input and learns the best features itself, as depicted in figure 2.2. In other words, the whole process is done end-to-end, eliminating the need for domain expertise.

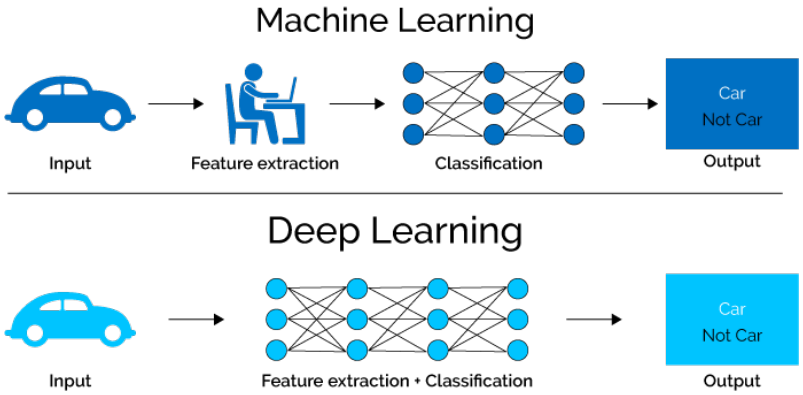


Figure 2.2: Traditional ML algorithms *versus* DL algorithms. In traditional ML techniques, features are identified and extracted by a domain expert, while DL algorithms learn high-level features directly from data. Extracted from [4].

These models have been shown to outperform previous state-of-the-art methods in a wide range of tasks, namely in the Computer Vision (CV) area. CV is a field focused on visual tasks such as image classification, object recognition, and image captioning. This area can be applied to many fields, namely medical.

2.1.1 Deep Neural Networks

A Deep Neural Network (DNN) consists of several processing layers able to learn and represent data with high levels of abstraction. The architecture behind these models is inspired by how the brain perceives and understands the information.

The basic processing units are the neurons, which can be grouped into the input layer, hidden layers, and output layer. The neurons are interconnected and pass the information to each other, mimicking the brain. Each connection between the neurons is associated with a weight, which is a measure of how much the output changes with the input. These weights are the learnable parameters of the network. Then, each neuron outputs a value by applying an activation function to the weighted sum of its inputs.

In supervised learning, to train the model there is a need for labeled data sets, so the network can learn by comparing its outputs with the expected ones. To do so, a function is created to describe how far the outputs were from the real ones — loss or cost function. The goal is to reduce the loss function by changing the weights between neurons using gradient descent to find the minimum of the function. In other words, by computing the derivative of the loss function with respect to the weights, the network can see in which direction the minimum is. The weights are then changed in small increments at each iteration in a process called backpropagation.

The following sections provide a brief description of a few global concepts regarding DNNs.

2.1.1.1 Weight Initialization

As mentioned above, the error is computed in each step with respect to the weights and propagated in reverse, i.e., from output to input. An adequate weight initialization is important to prevent gradients from exploding or vanishing. If this occurs, the network will take longer to converge or will not be able to do so [5].

The Xavier initialization was proposed by Xavier Glorot and Yoshua Bengio in 2010 [6]. Until then, the standard approach was initializing the weights from a uniform distribution in the range $[-1,1]$ and scaling by $1/\sqrt{n}$ in which n is the number of network input connections of a layer. However, this caused the gradients to be too small. This way, the Xavier or Glorot initialization sets each layer's weights within the range $\pm\frac{\sqrt{6}}{\sqrt{n_i+n_{i+1}}}$ in which n_i is the number of incoming connections and n_{i+1} the number of outgoing connections. They observed that the network maintained similar variances of its gradients across layers.

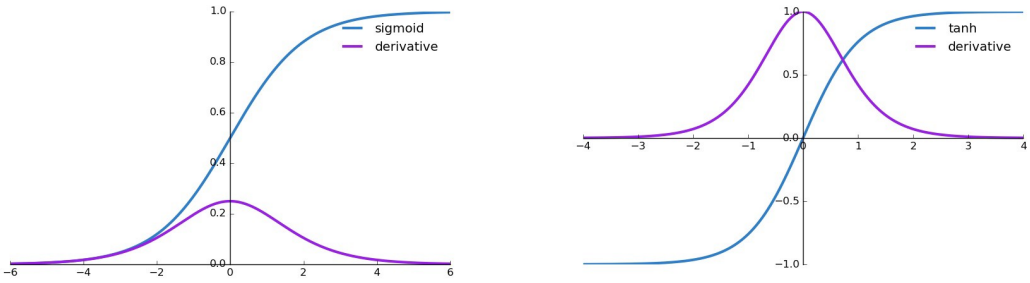
In 2015, Kaiming He *et al.* [7] proposed a new initialization approach for networks with asymmetric and non-linear activation functions, such as Rectified Linear Unit (ReLU)-like functions. The resulting weights are set within the range of $\pm\sqrt{\frac{6}{(1+a^2)\times fan\_in}}$.

2.1.1.2 Activation Functions

The activation functions define the output of a neuron given its input, indicating whether the neuron is activated or not. Hence, it follows a summary of some of the typical activation functions used in DNNs.

- **Sigmoid** is a step-like nonlinear function with a smooth gradient. The output of the function will always be in the range $[0,1]$. The main disadvantage is that towards either end of the function the gradient will be small.

- **Hyperbolic tangent** is similar to the sigmoid function but steeper, which translates into a stronger gradient, as depicted in figure 2.3. The output is within the range $[-1,1]$.



(a) Sigmoid (b) Hyperbolic tangent

Figure 2.3: Sigmoid and hyperbolic tangent functions with derivative. Extracted from [8].

- **ReLU** has a non-linear nature (figure 2.4). The output is within the range $[0, +\infty]$ and the gradient is zero for negative x values. It is less computationally expensive than the previous two. Currently, most DL algorithms use ReLU activation functions or variations of ReLU.

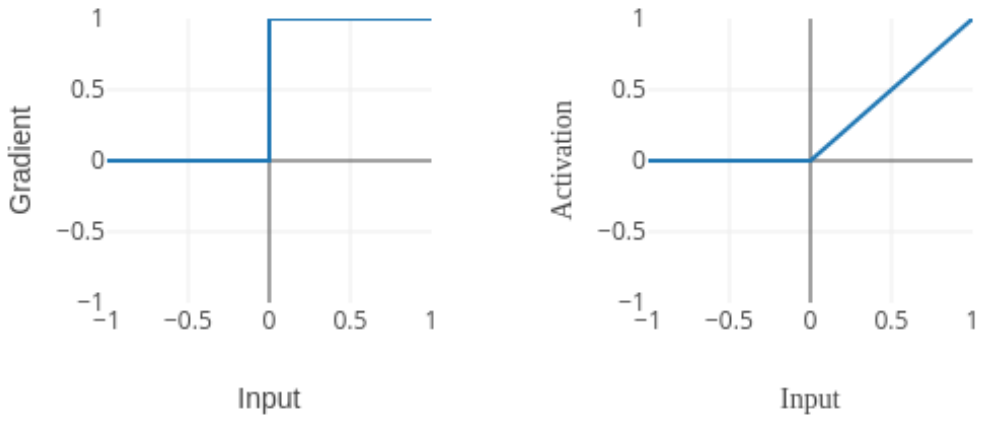


Figure 2.4: ReLU function with derivative. Extracted from [9].

2.1.1.3 Loss Function

In DL, the loss or cost function is a measure of how the model is performing. For classification tasks, the most common is the cross-entropy loss. If the predictions are far from the expected outputs, the loss will be high. In the training stage, the goal is to minimize this function to maximize performance.

2.1.1.4 Optimizers

Training a DL model is an iterative process in which the value of the cost function $J(\theta)$ is minimized by updating the learnable parameters θ. Optimization techniques can save time by reaching the minimum faster without a negative impact on performance.

While traditional gradient descent only updates the parameters after processing all of the training samples once, in *mini-batch gradient descent*, the training data is distributed in small mini-batches and the parameters begin being updated after processing the first mini-batch. This way, it is possible to train a model with a large amount of data even if it does not fit in the GPU memory. The mini-batch size becomes a hyper-parameter.

If the mini-batch size is set to one, meaning the weights are updated one sample at a time, it is called *stochastic gradient descent* (SGD) [10]. Although this algorithm is fast and simple to implement, the loss function fluctuates more and it is harder to converge to the exact minimum due to the frequent updates.

As mentioned above, high variance oscillations make it difficult to reach convergence. To handle this issue, *gradient descent with momentum* applies exponential smoothing to the gradient, leading to faster and stable convergence. This algorithm requires two new hyper-parameters: the learning rate (α) and the smoothing parameter (β). The learning rate is often called the most important hyper-parameter since it controls how much the model changes in response to the estimated loss, in each step.

Finally, *Adam* [11] stands for adaptive moment estimation and combines the advantages of AdaGrad [12] and RMSProp [13]. This method computes adaptive learning rates for each parameter based on estimates of lower-order moments. The authors determined that Adam outperforms other adaptive techniques in a variety of models and datasets.

2.1.1.5 Regularization

Training a DNN with a large number of parameters can lead to overfitting, i.e., the model fits the training data so well it negatively impacts the performance of the model on unseen data. Avoiding overfitting can be done through regularization techniques. *Dropout* [14] is a popular regularization technique in which neurons have a probability — dropout rate — of being ignored in a training step, as seen in figure 2.5.

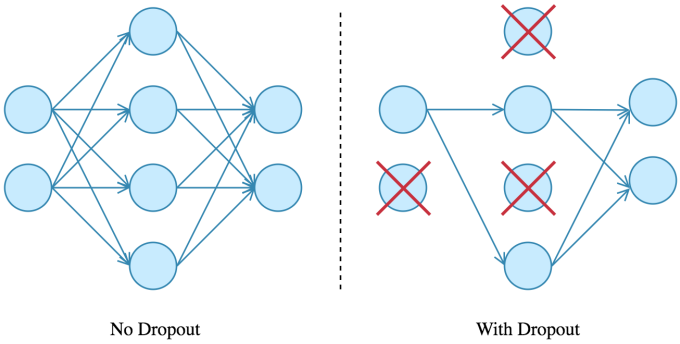


Figure 2.5: Dropout technique. Extracted from [15].

Another regularization technique is *Early Stopping* that consists of stopping the training when the loss function stops decreasing on the validation set. It is a simple yet effective and widely used method.

Having a large dataset is crucial to achieving the desired performance. To prevent overfitting and deal with the lack of data, one can perform *Data Augmentation*. This technique involves generating new data from the existing samples to increase the amount and diversity. Some of the most commonly used operations for data augmentation are as simple as translating, rotating, or cropping the images.

Another method for regularizing large convolutional neural networks is *Batch Normalization*, a method to normalize the inputs of a layer for each batch. It consists of computing the mean

and standard deviation over each batch of training data and use these statistics to standardize the inputs.

Finally, *Stochastic Pooling* can also be performed to cope with this problem. It consists of replacing the traditional deterministic pooling operations with stochastic ones, by randomly choosing the activation for each pooling region.

2.1.1.6 Transfer Learning

In DL, performance keeps increasing with the data size. However, large datasets are not always available. One way of dealing with data scarcity is transfer learning. Research on this topic was motivated by the human ability to apply knowledge learned previously to solve problems faster and/or better. Transfer learning techniques involve using data from a source domain to address a problem in a domain of interest, as depicted in Figure 2.6. This way, such techniques should be used when there is not enough target training data, and the source and target domains are similar.

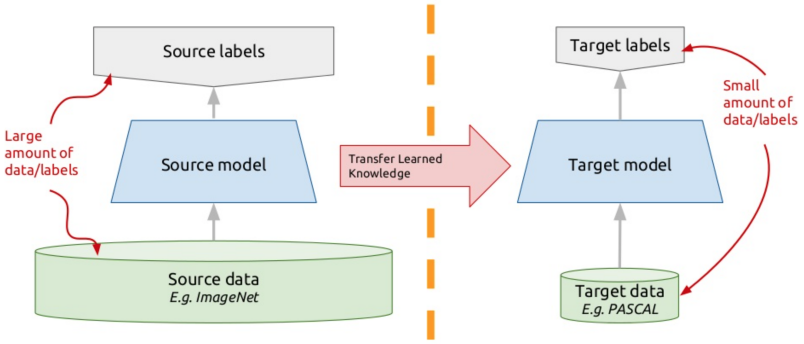


Figure 2.6: Basic flow of transfer learning. The knowledge is transferred from a source model to a target model. Extracted from [16].

Pan *et al.* [17] categorize transfer learning under three sub-settings:

- In an **inductive transfer learning** setting, the source task is different, but the source and target domains are the same. This way, labeled data in the target domain are required.
- In a **transductive transfer learning** setting, the source and target tasks are the same but the domains are different. Transductive transfer learning techniques are used when there is no labeled data available in the target domain.
- In the **unsupervised transfer learning** setting, the target task is different from but related to the source task. It is used when there is no labeled data available in any domain.

2.1.2 Convolutional Neural Networks

Traditional neural networks called MLPs are not fit to deal with images. MLPs use one perceptron for each input, which means for a single 224×224 pixel image with three channels there

are over 150.000 weights for each neuron that must be adjusted. Another issue is the incapacity of the model to preserve the spatial information once the image is flattened.

CNNs emerge as a type of feedforward DNN with applicability in visual understanding. A CNN takes an image as input, which can be described as a 3D matrix of numbers representing pixel values. Each unit in a CNN only depends on a region of the input space. This region is called the receptive field [18]. It is important to control the receptive field to ensure it covers relevant regions. As the image goes through the different layers of the network, the most meaningful features are extracted. In the first layers, low-level features such as edges, color, gradient orientation, or other primitive visual features are detected. Each upcoming layer learns to represent increasingly complex features by combining the ones in its preceding layers. A typical CNN architecture can be seen in figure 2.7.

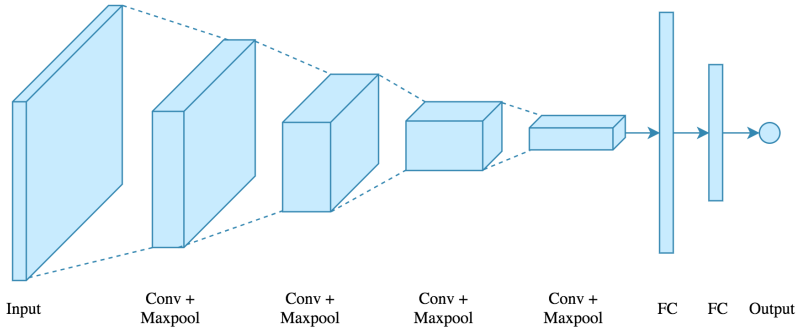


Figure 2.7: Typical CNN architecture. Extracted from [15].

2.1.2.1 Layers

As described above, a simple CNN is a sequence of layers. To build a CNN architecture there are essentially three types of layers — convolutional layers, pooling layers, and fully connected layers. Each one of these layers performs a different task.

- **Convolutional Layers** are the core of any CNN. The mathematical operation behind a convolutional layer consists of the dot product between the input and the filter or kernel. In general terms, the convolutional layer applies filters, composed of small kernels, to extract features. The first layers learn low-level features; as the layers stack-up, the architecture adapts to high-level features, with each layer outputting a set of feature maps. A particular attribute of the convolution is the preservation of the spatial relationship between pixels. Once a feature is detected, its approximate position relative to other features is preserved, instead of the exact position.

- **Pooling Layers** are used for subsampling or downsampling, which reduces the dimensionality of each map while still retaining important information. A pooling layer simply slides a window over its input, extracts the maximum (Max Pooling) or average (Average Pooling)

value of the region covered by the kernel, and propagates it to the next layer. This loss in dimensionality decreases the computational overhead for the subsequent layers.

- **Fully Connected Layers** take a 2D feature map as input and convert it into a 1D feature vector by creating linear combinations of the features. In multi-class classification tasks, these are generally followed by a softmax function to output the predicted classes. Usually, these layers simply aggregate information from the final feature maps and generate the classification.

2.1.2.2 Architectures

Although research on CNNs applied to visual tasks began in the late 1980s, it went largely unnoticed until the mid-2000s [19]. However, in the past few years, the developments in computing power and the growth of large public image repositories, such as ImageNet [20], resulted in innovative and improved CNN architectures. Some of the most prominent ones are described below, ordered by year of publication.

- **LeNet-5** is depicted as the first major contribution starting the era of CNNs [21]. Although LeNet is a rather simple network by modern standards, it is still one of the best known CNN architectures. Created by LeCun *et al.* [22], it was applied to handwritten digit recognition on the MNIST dataset. LeNet-5 consists of two stages of convolution-average pooling and two fully connected layers, followed by the output layer (Figure 2.8).

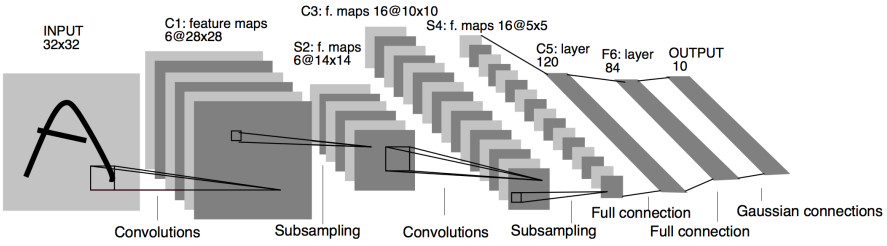


Figure 2.8: Architecture of LeNet-5. Extracted from [22].

- **AlexNet** [23] is similar to LeNet-5, but deeper — it comprises five convolutional layers and three fully-connected layers, with a total of 60 million parameters and 650,000 neurons. This paper was the first to implement ReLUs as activation functions for the hidden layers. In [21], AlexNet is considered the start of the age of CNNs used for ImageNet classification, opening the path for subsequent architectures. It achieved outstanding performance on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) of 2012, winning the competition by a large margin compared to the second-best entry.

- **GoogLeNet/Inception** [24] was inspired by AlexNet but implemented an inception module based on very small filters — 1×1 convolution to reduce dimensionality, 3×3 and 5×5

convolutions to deal with different scales, and finally a 3×3 max pooling for invariance — to dramatically reduce the number of parameters. With this approach, they were able to decrease the number of parameters from 60 million to 4 million.

- **VGG-16** [25] contributed mainly with an increasing depth. The model stacks more layers onto AlexNet and uses convolution filters with smaller (3×3) receptive fields, with a total of 138 million parameters. The obtained results proved the importance of depth in visual representation.

- **ResNet** [26] is a deep network. As mentioned above, depth is crucial for visual tasks. However, with increasing depth, accuracy gets saturated and then degrades. This paper popularized the use of shortcut connections as a solution to cope with the drop in accuracy with the increasing depth of the network.

- **DenseNet** [27] is a dense network in which each layer takes all preceding feature-maps as input and its feature-maps are used as input by all subsequent layers. Unlike traditional CNNs with L layers and L direct connections, this network introduces $L(L+1)/2$ connections. DenseNet helps with the vanishing-gradient problem, increases feature propagation and reuse, and reduces the number of learnable parameters.

The overall summary table of the described architectures can be seen below.

Table 2.1: ILSVRC competition CNN architecture models. Adapted from [28].

Year	**Architecture**	**Developed by**	**Error rates**	**No. of parameters**
1998	LeNet	Yann LeCun *et al.*	—	60 thousands
2012	AlexNet	Krizhevsky *et al.*	15.3%	60 million
2014	GoogLeNet	Szegedy *et al.* (Google)	6.7%	4 million
2014	VGGNet	Simonyan *et al.*	7.3%	138 million
2015	ResNet	Kaiming He *et al.*	3.6%	—
2017	DenseNet	Huang *et al.*	5.3%	0.8 million

Chapter 3

Explainable Artificial Intelligence

This Chapter is dedicated to a brief discussion of interpretability and explainability in ML, particularly in DNNs. First, the relevance of such concepts in real-world tasks is explored. Different methods that can be applied to explainability of DNNs are then described, followed by an analysis of how to measure the quality of an explanation, a subject on which there is still little consensus.

3.1 Concepts and Definitions

The adoption of complex ML models in an increasing number of tasks has brought a new challenge with it: how to understand and explain the root causes behind the models' predictions. In AI, this "*difficulty for the system to provide a suitable explanation for how it arrived at an answer*" is named the black-box problem [29]. Treating black-boxes as trustworthy models can lead to disastrous outcomes. On the other hand, choosing linear models over nonlinear models, despite their typically lower predictive power, to achieve higher interpretability is a trade-off that has to be taken into account (figure 3.1). Explainable Artificial Intelligence (XAI) emerges as a research field to cope with the lack of transparency of these models.

Explainability and interpretability are terms often used interchangeably. Although definitions are still not consensual, these concepts can be distinguished as follows. An interpretation is referred to as "*the mapping of an abstract concept into a domain that the human can make sense of*", while an explanation can be seen as "*the collection of features of the interpretable domain, that have contributed for a given example to produce a decision*" [1].

The numerous approaches to XAI can be classified in terms of when the explainability method is applied to the model. There are three stages of AI explainability: pre-model, in-model, and post-model explainability [31].

- *Pre-model* methods involve data visualization and exploratory data analysis. These methods help to understand and describe the data used to develop the model. This can be accomplished through methods such as K-Means or K-Nearest Neighbours (K-NN).

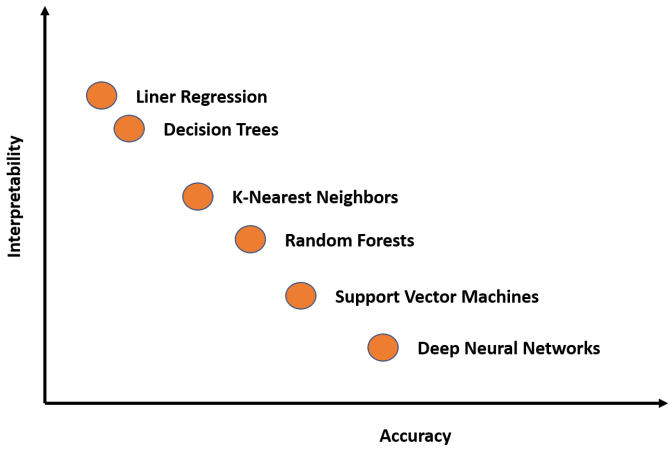


Figure 3.1: Correlation between accuracy and interpretability. Simpler and/or linear methods are less accurate but highly interpretable. Conversely, models that have a more complex nature produce more accurate results albeit being extremely difficult to interpret. Extracted from [30].

- *In-model* methods intend to develop inherently more explainable models. These can be rule-, per-feature-, case-, sparsity-, or monotonicity-based. Rule-based models include decision trees, rule lists, and rule sets. Although such models are interpretable, the increasing size can make the model too complex to monitor. Per-feature based methods contain linear models, generalized linear models, and generalized additive models. In case-based models, the results are explained using examples from a similar domain. However, it only works if there are representative examples. Sparsity-based models rely on the sparsity property. Sparsity is usually used to reduce the complexity of neural networks. By doing so, the model becomes more interpretable but performance often degrades. Finally, monotonicity-based models ensure that the model's monotonicity is preserved.
- *Post-model* or *post-hoc* explainability involves training the model and applying the explainability method after. This way, these methods extract explanations to describe pre-developed models, using perturbation mechanisms or proxy models, for instance.

The following sections will focus on the problem of interpreting complex ML models, namely DNNs, and the importance and benefits of these insights.

3.2 Relevance

As mentioned above, trusting these models without sanity checking can lead to tragic consequences. For instance, *Gender Shades* [32] is a study on the accuracy of AI-powered gender classification products. This study evaluated three commercial gender classification systems from the following companies: IBM, Microsoft, and Face++, on a new dataset balanced by gender and skin type. It concluded that all companies perform better on males than females, on lighter subjects than darker subjects, plus darker-skinned females are the most misclassified group, with an

error rate of up to 34.7% (figure 3.2). The significant accuracy discrepancy reported in this study demonstrates how important it is to ensure the model is fair and unbiased.

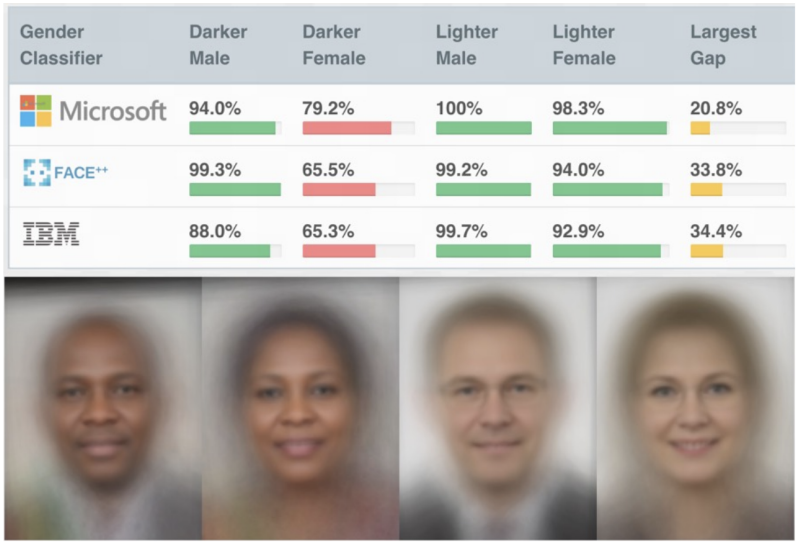


Figure 3.2: Gender classification performance as measured by the positive predictive value (PPV) of the three evaluated commercial gender classifiers on the PPB dataset. Extracted from [33].

In high-stakes domains, such as in medical diagnosis or self-driving cars, in which a single wrong prediction can be very harmful, the high predictive power of black-box models is not enough.

Besides the concerns about safety or nondiscrimination, a new European Union (EU) data protection law taking effect since 2018 addresses the risks of automated individual decision-making and profiling. The EU General Data Protection Regulation (GDPR) ensures that individuals have the right to an explanation for algorithmic decisions.

Doshi-Velez *et al.* [34] consider that an explanation is mandatory when there is an *incompleteness* in the problem representation leading to unquantified bias. Interpretability can be used to ensure that some requirements — fairness, privacy, reliability, robustness, causality, usability, and trust — are met.

3.3 Applications and Stakeholders

The adoption of DL across multiple real-world tasks increased the demand for XAI. Some of the most common use cases of XAI include data protection, transportation, healthcare, defense, and finance domains.

In the industry of self-driving cars, a wrong decision can have extremely harmful consequences, thus safety is the highest priority at every step of the research, development, and deployment processes. XAI is being explored to ensure autonomous vehicles operate safely, even when the system fails.

Healthcare is another field where explainability is important to increase trust in the model predictions. Decision support models are more valuable if they can present an explanation for the predictions they reach.

Finally, in the finance domain, a common scenario is when a person requests a loan and the financial institution rejects the loan application based on the output of a complex ML model that takes the customer's financial history and decides whether he/she is eligible to get the loan.

Following these use cases, a few key stakeholders can be defined, as depicted in figure 3.3. The data scientists or AI researchers must be analytical and have a solid understanding of the decision-making process of a model. They can collaborate with business teams/non-technical users to adapt the explanations for commercial purposes. The consumers will trust a business if they are provided with full transparency. Finally, a model must be compliant with the legislation. Policymakers have a critical role in providing guidelines and ensuring the protection of consumers.

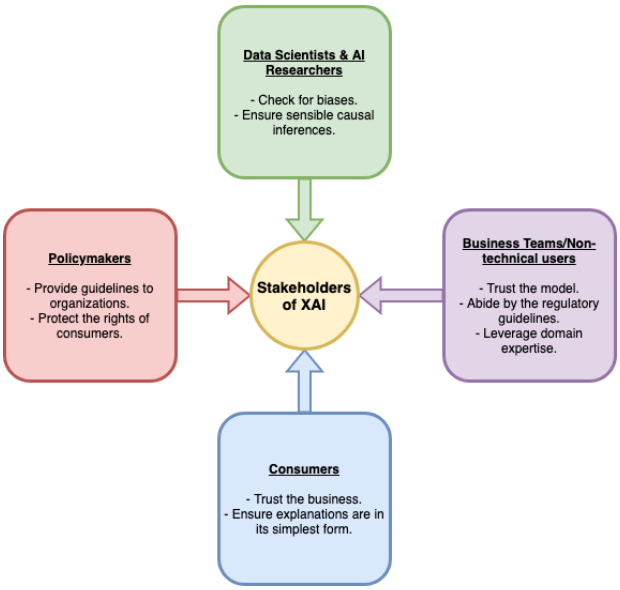


Figure 3.3: XAI stakeholders and their attributes. Extracted from [35].

3.4 Explainability of Deep Neural Networks

One of the most recent and promising approaches for interpreting DL models is Testing with Concept Activation Vectors (TCAV) [36]. Kim *et al.* introduce Concept Activation Vectors (CAVs) as an interpretation of the network's internal state in terms of human-interpretable concepts. CAVs are determined by training a linear classifier between samples representative of a concept of interest and counterexamples randomly chosen. The CAV is the vector orthogonal to the decision boundary and can be used to describe the importance of a user-defined concept for a prediction.

Essentially, TCAV learns *concepts* from examples, as depicted in figure 3.4. This method differs from other state-of-the-art methods as it shows the importance of high-level concepts for a prediction, bridging the gap between man and machine.

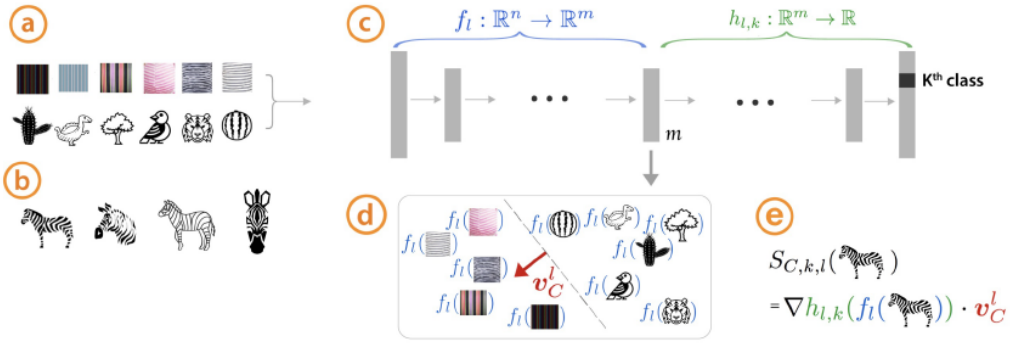


Figure 3.4: TCAV pipeline: Given a user-defined set of examples for a concept (e.g. "striped"), and random examples (a), labeled training-data examples for the studied class — zebras (b), and a trained network (c), TCAV learns to quantify the importance of the concept for that class. A linear classifier is trained to distinguish between the activations produced by a concept's examples and examples in any layer (d). The CAV is the vector orthogonal to the classification boundary (red arrow). The conceptual sensitivity is estimated using the directional derivative (e). Extracted from [36].

Montavon *et al.* [1] also address the problem of interpreting the modeled concepts and explaining individual decisions made by the model in their work. They review three methods for explaining DNNs with concern to the function $f(x)$ implemented by the model, which are briefly described in the following sections. All three methods produce an explanation in the shape of a heatmap, in which red and blue indicate positive and negative relevance scores, respectively (figure 3.5). From a pixel-wise heatmap, one can produce a region-wise heatmap by pooling the relevance scores, for instance.

3.4.1 Sensitivity Analysis

Sensitivity Analysis (SA) [37, 38, 39] involves evaluating the model's gradient locally. Thus, the relevance score evaluated at point x can be defined as $R_i(x) = (\frac{\delta f}{\delta x_i})^2$. The most relevant features are the ones that the output is most sensitive to.

3.4.2 Simple Taylor Decomposition

Simple Taylor Decomposition [40, 41] decomposes the function $f(x)$ as a sum of relevance scores. These are determined based on the terms of a first-order Taylor expansion of the function at a root point $\tilde{x}$ for which $f(\tilde{x}) = 0$.

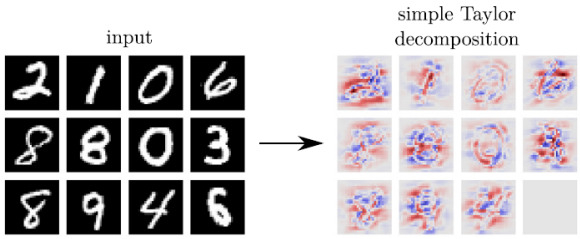


Figure 3.5: Resulting heatmap of a simple Taylor decomposition applied to a CNN trained on the MNIST dataset. Extracted from [35].

3.4.3 Backward Propagation Techniques

Backward Propagation Techniques, as the name suggests, start from the output of the network and map the prediction through preceding layers until the input is reached. These techniques include *Layer-Wise Relevance Propagation* (LRP) [41], in which each neuron receives a portion of the output and redistributes it backward in equal amounts until the input is assigned a relevance score, as seen in figure 3.6.

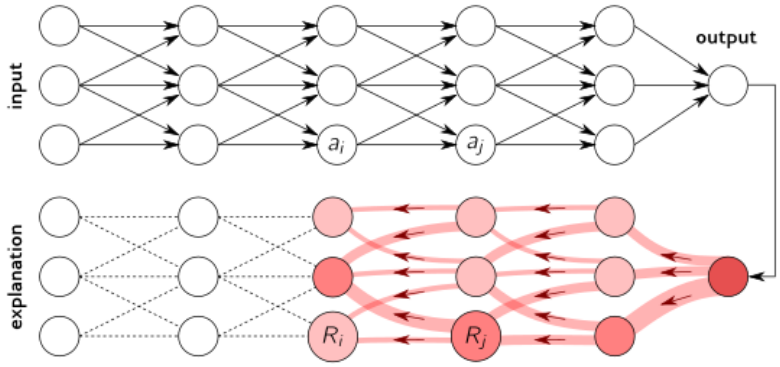


Figure 3.6: Diagram of the LRP procedure. Red arrows indicate the relevance propagation flow. Extracted from [1].

3.5 Other Methods

Local Interpretable Model-Agnostic Explanations (LIME) [42], Shapley Additive Explanations (SHAP) [43] and Model Agnostic Supervised Local Explanations (MAPLE) [44] are examples of alternative methods for model interpretability.

LIME, as the name suggests, is model-agnostic, meaning it can be applied to *any* kind of classifier. This explanation technique involves learning an interpretable model locally by choosing a representative data sample and observing the impact of adjusting the values of the features on

the output. This way, LIME outputs a list of explanations that translate the contribution of each feature for the given prediction.

Lundberg *et al.* propose SHAP values as a measure of feature importance. SHAP outputs an importance value to each feature given a particular prediction. It is a local technique, similar to LIME.

MAPLE differs from the previous because it is effective both as an explanation system and a predictive model. The authors claim that MAPLE is at least as accurate as random forests while producing better explanations than LIME.

3.6 Evaluation

There is still little consensus on how to evaluate or measure the quality of an explanation. The quality of an explanation has a subjective nature and depends on many task-related aspects: whether it is a global or local explanation, the source of concern and/or severity of incompleteness, the time constraints, and the nature of the user expertise [34].

When generating an explanation, one must consider how long the user can afford to spend to understand the explanation, how experienced the user is in the task, or what level of sophistication he/she expects from the explanation.

Montavon *et al.* [1] tackle the issue of evaluating the quality of an explanation, introducing the following proxy functions.

- *Explanation Continuity* relies on the assumption that two equivalent data points have similar explanations.
- *Explanation Selectivity* measures how fast the function value $f(x)$ decreases when removing the features with the highest relevance scores.

Additionally, Finale Doshi-Velez and Been Kim propose three evaluation approaches for interpretability in their work [34], which are briefly described in the following sections.

3.6.1 Application-Grounded Evaluation

Application-Grounded Evaluation involves conducting human experiments within a real application. In other words, it evaluates the quality of an explanation in the context of the end-task. For instance, if the model was designed to help with the medical diagnosis of a specific condition, determining whether it works consists of evaluating it in the context of doctors diagnosing. Although it is a simple and intuitive method it is highly expensive and requires domain experts.

3.6.2 Human-Grounded Evaluation

Human-Grounded Evaluation is similar to the previous approach — it involves conducting simpler human-subject experiments while retaining the essence of the target application. This method is appropriate when one wishes to test more generic concepts of an explanation.

3.6.3 Functionally-Grounded Evaluation

Functionally-Grounded Evaluation requires no human experiments, unlike the previous two, relying on proxy tasks instead. Accordingly, it is appropriate when there is a class of models or regularizers that have already been validated, when a method is not mature enough, or when human-subject experiments are unethical. It is also less expensive than previous methods. On the other hand, proxy tasks are not accurate measures of the explainability.

Chapter 4

Automatic Generation of Text from Images

This chapter focuses on a comprehensive review of the existing literature on automatic image captioning. First, some fundamental concepts on NLG, one of the pillars for automatic generation of text from images, are introduced. Then, the evolution of language generation from simpler methods, such as Markov chains, to the established state-of-the-art approaches in sequence modeling tasks, is briefly discussed. The subsequent sections focus on image captioning methods, namely medical imaging report generation, which differs from traditional image captioning where usually a single sentence is required, as well as the datasets and metrics commonly used to evaluate the quality of the generated text.

4.1 Natural Language Generation

NLG is the process of turning data (structured or not) into meaningful text, understandable to humans. Instead of offering information structured in tables, charts, or graphs (which requires interpretation), NLG can provide it via natural language. NLG can be applied across a variety of tasks, namely image captioning, one of its most impressive research areas. Given an image, the goal is to understand its content and generate a sentence describing what the image consists of. So, unlike other NLG tasks, the output is based on a non-linguistic representation.

Additionally, the expected evolution of this field is to provide efficient communication between humans and computers more naturally. For instance, NLG combined with Natural Language Processing (NLP) is already at the core of automated chats and assistants [45]. The ability to turn data into clear, natural language has changed the way companies interact with their data, and it is a process that is in constant growth by itself.

NLG models usually start by transforming the text into a more appropriate representation. For words to be processed by a model, they need to be converted to a numeric representation that captures semantic and syntactic relationships. Conceptually, if two words are similar, their vector representations should be similar as well. That being said, words are usually represented by

embeddings. These can be trained from scratch alongside the model or pre-trained using popular methods such as Word2Vec [46] or GloVe [47], for instance. Pre-training word embeddings on a large corpus of text can be particularly interesting when using datasets with a considerable volume of rare words, like in most real-world problems.

4.1.1 Evolution

A sentence can be thought of as a sequence of words. That being said, models capable of processing sequential data are at the core of NLG. The following sections provide an overview of the evolution of such models.

4.1.1.1 Markov Chain

Markov chain [48], named after the Russian mathematician Andrey Markov, is one of the most common algorithms used in language generation. Markov chains are based on the stochastic process in which the next state of the system depends only on the present state (figure 4.1), not on the preceding ones. With Markov chains, the next word in a sentence is predicted given the previous word only. In that sense, the Markov chain was used in earlier versions of the smartphone keyboard. However, this model has limited applicability because it does not account for the context of the sentence.

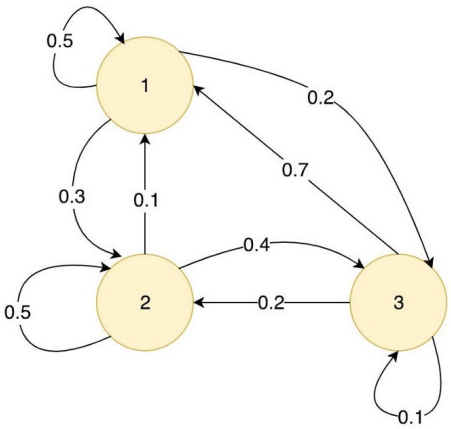


Figure 4.1: Markov process showing the probability of the transition from one state to another. The probability of reaching a state from any state is one. Extracted from [49].

4.1.1.2 Recurrent Neural Network

RNNs are a type of ANNs capable of dealing with sequential data such as sentences. The primary research work on RNNs was introduced by Hopfield in 1982. Hopfield [50] proposed a model with highly interconnected neurons. However, his work was not designed to process sequences of patterns. The first network of this kind was introduced by Jordan in 1986. The

Jordan network [51] consists of connections between the output in one state and the hidden cell in the following state.

Unlike traditional feedforward networks, RNNs have sequential memory, making it easier to recognize sequence patterns. This is achieved through the hidden state, which is a representation of previous inputs, allowing the information to persist throughout the sequence (figure 4.2). In each iteration, prior information is stored and used to generate the word distribution over the dictionary (a mapping between the words and integers). Then, the word with the highest score is selected, and so on. This internal memory makes RNNs ideal language models.

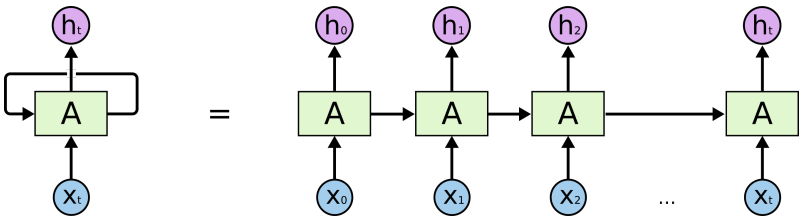


Figure 4.2: RNN model diagram. A RNN can be seen as a chain of multiple copies of the same network, each passing the information to its successor, allowing information to persist. Extracted from [52].

Although RNNs have been widely used for sequence learning tasks, they suffer from vanishing gradient problems. Learning long-term dependencies with stochastic gradient descent is hard to accomplish [53]. As the length of the sequence increases, the gap between the relevant information and the point where it is needed becomes wider, making it difficult to connect the information.

Backpropagation through time (BPTT) [54] is an extension of the backpropagation method for RNNs. In BPTT, the error is propagated back in time for a specific number of time steps. As can be seen in figure 4.3, BPTT training provides significant improvements compared to standard backpropagation.

4.1.1.3 Long Short-Term Memory

As mentioned above, vanilla RNNs are unable to capture long-range dependencies. To handle this limitation, a new model was proposed. LSTMs [56] are a variant of RNNs that share the chain-like nature but with internal mechanisms that can store information for longer periods.

The cell state (the top horizontal line in figure 4.4) is where the information flows along. An LSTM has three gates, composed of sigmoid layers and pointwise operations, each with a particular task. These gates are what allow the LSTM to regulate the flow of information in and out of the cell state.

Each decoding step can be formally described by equations 4.1 to 4.6. First, the forget gate, f_t, decides what information from prior steps can be removed from the cell state by looking at the previous hidden state and the current input (4.1). Next, the input gate, i_t, decides which information from the current state can be stored (4.2) and a hyperbolic tangent layer creates a vector of new candidates to be added to the cell state (4.3). At this point, the cell state is ready to

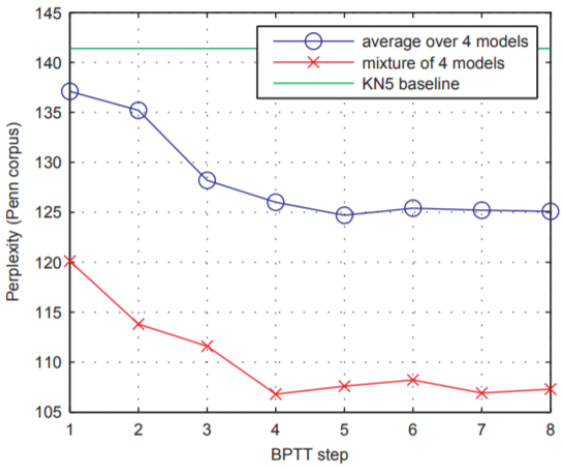


Figure 4.3: Effect of BPTT training. Each curve represents a different training procedure. The x axis corresponds to the number of time steps ($BPTT = 1$ coincides with standard backpropagation). The y axis can be understood as a measure of the uncertainty of the model. This figure shows the importance of the number of time steps to obtain lower perplexity. Extracted from [55].

be updated (4.4). Finally, the output will be a filtered version of the latter. To do so, the output gate, o_t, decides what parts of the cell state can be output (4.5), determining the next hidden state (4.6) [58].

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{4.1}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{4.2}$$
$$g_t = \tanh(W_g[h_{t-1}, x_t] + b_g) \tag{4.3}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{4.4}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{4.5}$$
$$h_t = o_t \odot \tanh(c_t) \tag{4.6}$$

where h_t, c_t, and x_t are the hidden state, cell state, and input at time t, h_{t-1}, c_{t-1}, and x_{t-1} are the hidden state, cell state, and input at time $t-1$, and i_t, f_t, g_t, and o_t are the input, forget, cell, and output gates of the LSTM, respectively. $W_\bullet$ and $b_\bullet$ are learned weight matrices and biases. σ is the Sigmoid function and $\odot$ is the Hadamard, or element-wise, product.

In theory, LSTMs solve the issue of vanishing gradients. Nevertheless, there is still a limitation concerning the length of sequences an LSTM can handle.

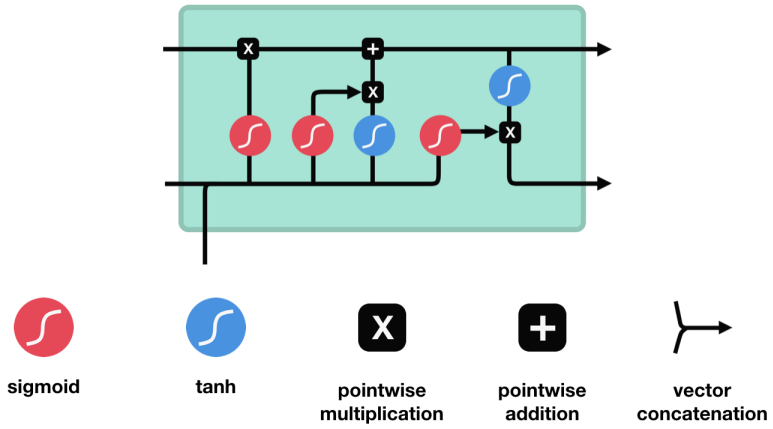


Figure 4.4: LSTM cell and its operations. Extracted from [57].

4.1.1.4 Gated Recurrent Unit

The Gated Recurrent Unit (GRU), proposed by Cho *et al.* [59], is similar to an LSTM but consists of only two gates — an update gate and a reset gate —, as can be seen in figure 4.5. The update gate determines what information to keep while the reset gate decides how much past information can be forgotten. The information is stored in the hidden state.

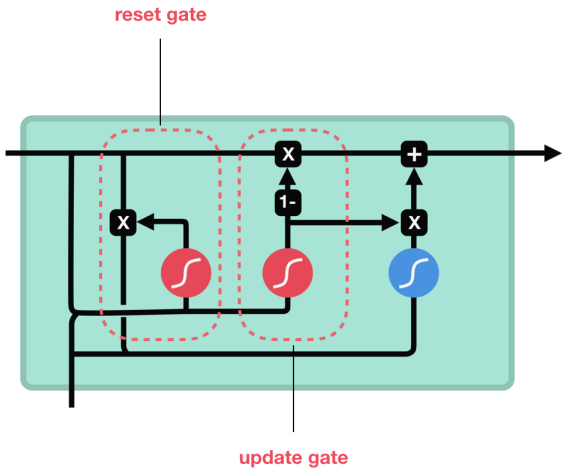


Figure 4.5: GRU cell. Extracted from [57].

Each decoding step can be described by equations 4.7 to 4.10:

$$r_t = \sigma(W_r[h_{t-1}, x_t]) \tag{4.7}$$
$$z_t = \sigma(W_z[h_{t-1}, x_t]) \tag{4.8}$$
$$n_t = \tanh(W[r_t \odot h_{t-1}, x_t]) \tag{4.9}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot n_t \tag{4.10}$$

where h_t and x_t are the hidden state and input at time t, h_{t-1} is the hidden state at time $t-1$, and r_t, z_t, n_t are the reset, update and new gates, respectively. $W_\bullet$ and $b_\bullet$ are learned weight matrices and biases. σ is the Sigmoid function and $\odot$ is the Hadamard, or element-wise, product.

4.1.1.5 Transformer

Until 2017, RNNs, LSTMs, and GRUs were the dominating architectures for sequence-to-sequence tasks such as language modeling and machine translation. However, RNNs are unable to capture long-range dependencies and although LSTMs possess long-term memory, learning such dependencies remains a major challenge.

In the field of machine translation, considering a decoder that is presumed to translate potentially long paragraphs based on the last hidden state from the encoder, it is unreasonable to assume that it is possible to encode all information into a single vector and the decoder will be able to produce a proper translation based on such information. To tackle this issue, a novel self-attention-based architecture was proposed by Vaswani *et al.* — the transformer [60].

Self-attention is a mechanism that relates different positions within a sequence to determine a new representation of the sequence. For instance, in the sentence *"Peter can be annoying but he is a great brother"*, the words *"Peter"*, *"he"*, and *"brother"* all refer to the same person. When processing the word *"he"*, self-attention allows the model to link this word to *"Peter"*. That is, given the input vectors $x_1, x_2, ..., x_n$ and the corresponding output vectors $y_1, y_2, ..., y_n$, the basic operation behind self-attention can be seen as a weighted average over the input vectors:

$$y_i = \sum_j w_{ij} x_j \tag{4.11}$$

Each input vector x_i is used in three different ways: as a query vector when compared to all other vectors to compute the weights for its output y_i; as a key vector when compared to all other vectors to compute the weights for the output of the j^{th} vector y_j; and as a value vector when used as a part of the weighted sum to compute each output vector. So, the output is a weighted sum of the values in which the weight assigned to each value is a function of the query and key vectors, as seen in figure 4.6.

Overall, the transformer involves a stack of encoder and decoder layers. It also performs multi-head attention by running several attention functions in parallel. The architecture is depicted in figure 4.7.

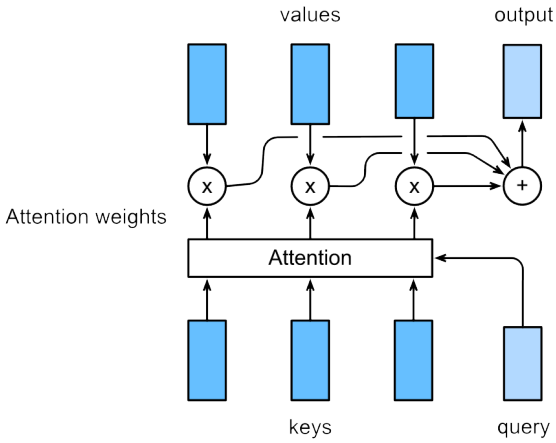


Figure 4.6: Self-attention block. The output is a weighted sum of the values, where each weight is a function of the query and the keys. Extracted from [61].

4.2 Image Captioning in Deep Learning

As previously mentioned, while a traditional language model predicts the probability of the next word in the sequence given the previous words, it can be extended into a caption generator by conditioning predictions on image features.

The first approach to image captioning using neural networks was proposed by Kiros *et al.* [62]. In the earliest works, the most common approach for image captioning consisted of using a CNN for feature extraction, to obtain a single fixed-length representation of the image, followed by an RNN to generate captions. The visual feature extraction can be thought of as a dimensionality reduction while preserving the most relevant information. Then, the networks communicate through this latent space vector. More recently, attention modeling has also been shown to achieve competitive performance.

The existing techniques can be grouped into three main categories: template-based image captioning, retrieval-based image captioning, and novel image caption generation. Template-based captioning usually consists of a simple gap-filling approach based on the image content in terms of objects, their attributes and the relationships that hold between them. The generated captions are simple and grammatically correct but lack flexibility, diversity, and scalability. Retrieval-based captioning is based on the assumption that similar images have alike captions. Most methods fall into the latter — novel image captioning —, which involves analyzing the visual content of the image and use it to generate captions with a language model. These can also be divided into other categories according to the following criteria: feature mapping, type of learning, number of captions, architecture, and others [63].

An overall taxonomy of the aforementioned DL-based image captioning methods is depicted in figure 4.8.

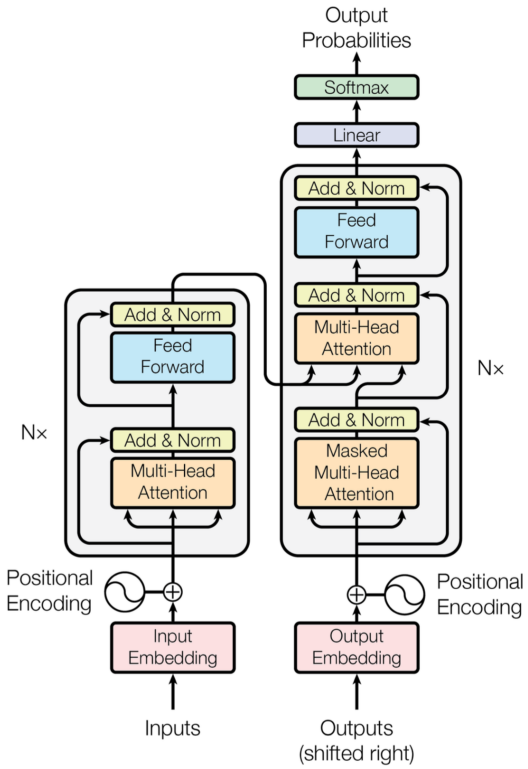


Figure 4.7: The transformer model architecture. First, the positional encoding remembers in which order the elements of the sequence were fed into the model (unlike RNNs, transformers do not require the sequential data to be processed in order). Both the encoder (left) and decoder (right) are composed of stacked layers, N_x. Each layer of the encoder has two sub-layers — the first is a multi-head attention mechanism and the second is a position-wise fully connected feedforward network. There is a residual connection around each sub-layer to avoid the vanishing gradient problem. The decoder is rather similar to the encoder. However, each decoder layer adds a third sub-layer that performs multi-head attention over the output of the encoder layer. Lastly, a linear transformation and a softmax function are used to convert the decoder output to the predicted word probability. Extracted from [60].

4.2.1 Taxonomy of Deep Learning-based Image Captioning

4.2.1.1 Feature Mapping

In visual space-based methods, the image features and the caption are independent, while in multimodal space, the image and corresponding caption are mapped into a common space that is then passed to the decoder [62]. A large amount of DL-based image captioning methods use visual space and are further explored in the following sections.

4.2.1.2 Type of Learning

In supervised learning, the training data is labeled. Conversely, unsupervised learning deals with unlabeled data. There are several supervised-learning-based image captioning methods,

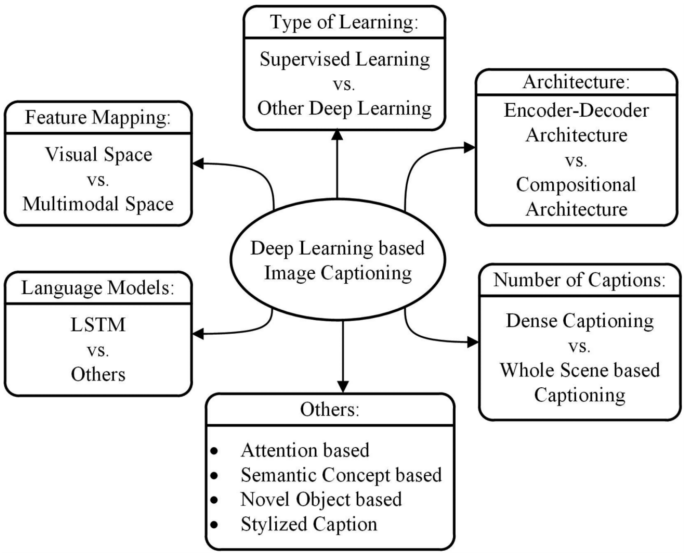


Figure 4.8: Overall taxonomy of DL-based image captioning. Extracted from [63].

which can be classified into encoder-decoder architecture, compositional architecture, attention-based, semantic concept-based, stylized captions, novel object-based, and dense image captioning. However, it is often difficult to come by annotated data, thus being important to focus more on unsupervised-learning-based techniques. The methods in the latter use a CNN- and RNN-based combined network to generate captions, followed by another one that evaluates such captions and sends feedback to the preceding network.

4.2.1.3 Number of Captions

In dense captioning [64], the model produces region-wise captions. A dense localization layer determines regions of interest in the image and a CNN obtains the region-based image features. Finally, an LSTM language model is used to generate captions for every region. Region-based descriptions are more detailed than global image descriptions generated by whole-scene-based methods. Encoder-decoder architecture, compositional architecture, attention-based, semantic-concept-based, stylized captions, novel object-based image captioning, and other DL-based image captioning methods are all whole-scene-based methods.

4.2.1.4 Architecture

The encoder-decoder architecture involves extracting the image features from the hidden layers of a CNN and using them as an input to the LSTM language model to generate a sentence, as described in figure 4.9.

When generating image captions, image features are included in the initial state of an LSTM so that the next words are generated based on that information. However, since image information is only fed at the beginning, it may vanish. Because of this, LSTMs struggle with the generation

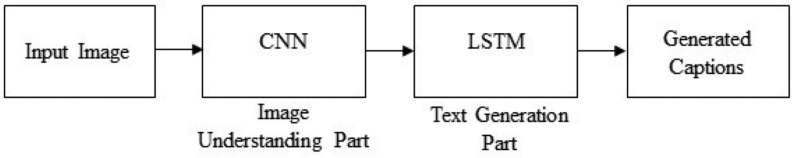


Figure 4.9: Block-diagram of typical encoder-decoder architecture-based image captioning. The encoder stage receives an image as input and encodes it. The encoded image is then fed to the decoder that outputs the captions. Extracted from [63].

of long-length sentences. To handle this issue, Jia *et al.* [65] proposed a guided LSTM (gLSTM). This model is an extension of the original LSTM that can generate longer sentences, by adding global information to each gate and cell state of the LSTM.

Compositional architecture-based methods involve independent building blocks. First, a CNN obtains the image features, which are converted into visual concepts. These visual concepts are then fed to the language model. The latter generates multiple captions using the information previously gathered and these are ranked using a deep multimodal similarity model.

4.2.1.5 Others

Methods unrelated to other models include attention-based, semantic-concept-based, novel object-based methods, and stylized captions.

- Attention-based mechanisms are gaining popularity because they can dynamically focus on different regions of interest within the given image in each step of the process, i.e. while the sentence is being generated. Xu *et al.* [66] were the first to propose a mechanism of this kind. The attention model uses features from the first convolutional layers to preserve the information the most. This way, the decoder knows where to focus when generating each word of the sentence.

- Semantic-concept-based methods include extracting semantic concepts from the image using a CNN-based encoder and adding them to the hidden states of the language model. Similarly to other mechanisms, the image features are extracted using the same encoder and passed as input to the LSTM-based language model. You *et al.* [67] introduced a semantic-attention-based model that learns to attend to semantic concepts. While the model proposed by Xu *et al.* [66] worked on a fixed spatial location, this method can work on any location of the image.

- Novel-object-based methods can generate descriptions of unseen objects by training a lexical classifier and a language model on unpaired image data and unpaired text data and combining these with the model trained on paired image-caption data. Yao *et al.* [68] proposed a method that uses a separate object recognition dataset to learn to recognize unseen objects and integrate them with the LSTM to generate captions.

- Stylized captions consider the stylized part of the text. The methods in this category are coupled with a text corpus to extract various stylized concepts from the training data. This way, the generated descriptions are more expressive and attractive than factual descriptions. Gan *et al.* [69] introduced StyleNet, a method capable of producing attractive captions with a specific style.

4.2.1.6 Language Models

LSTM networks are an extension of the RNN with a cell memory that can hold information for longer periods. However, these methods require significant storage to handle long-term dependencies.

4.3 Evaluation Metrics

Kougia *et al.* [70] state that the most common evaluation metrics in medical image captioning are Bilingual Evaluation Understudy (BLEU) [71], Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [72], and Metric for Evaluation of Translation with Explicit Ordering (METEOR) [73]. However, BLEU and ROUGE have been shown to correlate weakly with human assessments of quality [74, 75, 76, 77]. More recently, two additional metrics for image captioning evaluation were proposed: Consensus-based Image Description Evaluation (CIDEr) [78] and Semantic Propositional Image Caption Evaluation (SPICE) [79].

4.3.1 BLEU

BLEU is the most common metric, originally developed for machine translation evaluation. It compares the generated text segments with a set of reference texts and computes a score for each of them. To estimate the overall quality of the generated text, the scores are averaged. However, the final score depends on the number of references and the size of the generated text. A modified precision metric based on n-grams was later introduced by the same authors. It measures word n-gram overlap between the generated and the reference descriptions. The matches are position-independent and the more they are, the better the candidate translation is [71]. The score ranges from zero to one, with the latter meaning a perfect match. This way, BLEU can be used in image captioning to evaluate the quality of the generated caption.

Although BLEU is a pioneer in evaluating the quality of a generated text, there has been considerable criticism of this score. BLEU has a rather poor correlation with human judgment — it does not consider meaning, implying that a difference in a function word is as penalized as a difference in a content word. Furthermore, synonyms are not valid matches. On the other hand, this score offers many benefits — it is inexpensive to calculate, easy to understand, and language-independent — which is why it is still one of the most popular metrics for evaluating sequence to sequence tasks.

4.3.2 ROUGE

ROUGE is commonly used for measuring the quality of text summaries, by comparing the word sequences, word pairs, and n-grams with a set of references. There are five variants of ROUGE, according to whether the focus is on n-grams co-occurrence (ROUGE-N), the longest common sub-sequence (ROUGE-L), weighted longest common sub-sequence (ROUGE-W), or skip-bigram co-occurrence (ROUGE-S). Lastly, ROUGE-SU is an extension of ROUGE-S with the unigram as a counting unit.

4.3.3 METEOR

METEOR was designed to tackle the weaknesses in BLEU, such as the lack of recall, the use of higher-order n-grams as a measure of the level of grammaticality, the lack of explicit word-to-word matching, and the use of geometric averaging of n-grams, which can lead to senseless sentence-level scores. METEOR takes word segments and compares them with the reference texts. It takes into account both stemming (Porter stemmer) and synonymy (WordNet) for matching, making a better correlation with the human assessment.

4.3.4 CIDEr

CIDEr was created to evaluate image descriptions by comparing the generated sentence against a set of ground truth sentences written by humans. All words are stemmed and the sentences represented by n-grams. CIDEr then measures how often n-grams from the generated sentence are contained in the reference ones. Finally, common n-grams are given lower weight, since they are likely to be less informative. To do so, CIDEr uses a term frequency-inverse document frequency (TF-IDF) to weigh each n-gram.

4.3.5 SPICE

SPICE is a caption evaluation metric that outperforms all previous ones in terms of agreement with human judgment. It uses a graph-based semantic representation of the text, called a scene graph, to extract information on objects, attributes and/or relationships. This way, SPICE can learn things such as which image captioning method best understands colors, for instance.

Although SPICE and CIDEr correlate with human judgment better than the previous metrics, it is difficult to optimize them. This way, Liu *et al.* [80] proposed a combination of SPICE and CIDEr, named SPIDEr, that uses a policy gradient optimization.

4.4 Medical Image Captioning

Medical image captioning methods are designed to assist physicians to focus on regions of interest or describe findings within an image. These methods are more complex than classical image captioning ones because the annotations are usually longer.

Despite its importance, there are still few resources available, namely appropriate datasets. Moreover, most methods use image captioning datasets containing generic images.

Schlegl *et al.* [81] used semantic concepts extracted from clinical reports linked to the image data to train a CNN. Since medical reports usually list observations coupled with their location, the model learned the relationship between such concepts and their location. The authors evaluated the method on high-resolution images of the retina and showed that the model links semantic concepts to the image content, achieving overall better classification accuracy.

Kisilev *et al.* [82] proposed a multi-task CNN-based architecture for detection and semantic description of lesions. First, the regions of interest were detected and ranked. Then only the regions with the highest scores were fed into subsequent layers, which generated the semantic description.

Shin *et al.* [83] were the first to implement a CNN RNN encoder-decoder approach for medical image captioning. They used the IU X-Ray dataset and the image annotations to extract disease-related terms. The most frequent terms were used as labels to train a CNN. However, this disease image label mining only accounts for 40% of the entire dataset. They also adopted regularization techniques — batch normalization and data dropout — to handle the class imbalance. Then, RNNs generated the descriptions based on the image features.

Zhang *et al.* [84] introduced an attention-based model in biomedical image captioning. MD-Net used a ResNet to generate the image representation, which is then used as the starting hidden state of an LSTM-based decoder, coupled with an attention mechanism. They used the BCIDR dataset with bladder cancer images and the respective reports. Later, Zhang *et al.* [85] introduced a multimodal network, TandemNet, based on a dual-attention model for accurate image prediction. The network takes an image and an optional report as input. Each report describes multiple types of features structured into sentences. The model uses an LSTM to extract the hidden state of each feature description, obtaining a final text representation matrix, S. This way, the network relies on knowledge from different modalities to provide both visually and semantically interpretable diagnostic predictions through natural language descriptions and attention visualization. The architecture of the TandemNet can be seen in figure 4.10.

The authors propose two techniques for the prediction module:

- *Visual skip-connection* in which the visual feature skips the dual-attention model. In back-propagation, this connection passes the gradients directly to the loss layer of the CNN preventing possible vanishing gradient issues in the dual-attention model from influencing the CNN training.

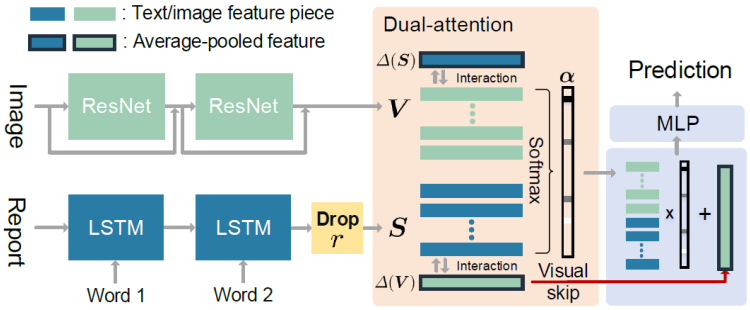


Figure 4.10: Illustration of the TandemNet, a multimodal network that jointly learns from medical images and reports. The dual-attention model enables the interaction between the visual and semantic representations in order to provide accurate predictions. Extracted from [85].

- *Stochastic modality adaptation* defined a drop rate, r, as the probability to remove the text representation to deal with the absence of text without a negative impact on the performance.

Overall, the model performs better when text is provided. However, when the drop rate is low, the model tends to overuse the text information and is unable to generalize when no text is provided. On the other hand, a high drop rate significantly decreases the accuracy with or without text.

Jing *et al.* [86] brought three main contributions. First, they proposed a multi-task framework that treats the prediction of relevant tags from the visual features as a multi-label classification (MLC) task, besides generating the medical imaging reports.

Next, they emphasize that a single-layer LSTM is not capable of modeling long, descriptive image paragraphs — a complete report is usually long, containing multiple sentences. Similarly to Krause *et al.* [87], they built a hierarchical LSTM, consisting of a sentence-level single-layer LSTM followed by a word-level two-layer LSTM. The first produces high-level topics and determines the number of sentences to generate, while the second generates the words of a sentence given a topic vector. In the end, all sentences are concatenated to generate the paragraph. They compare the performance of the model with a simpler CNN-LSTM method proposed by Vinyals *et al.* [88], showing the effectiveness of the hierarchical approach.

Finally, they compare the performance of the model using no attention, visual attention only, semantic attention only, and both visual and semantic attention. With this experiment, they proved visual attention or semantic attention alone are not sufficient and adopted a co-attention mechanism, using both visual and semantic information, to detect and focus on regions containing abnormalities.

An illustration of the proposed model can be seen in figure 4.11. Similarly to Shin *et al.* [83], results are reported on the IU X-Ray and PEIR Gross datasets.

Yuan *et al.* [89] also used an hierarchical LSTM, i.e. the combination of sentence- and word-level LSTMs, with a few contributions. First, the encoder was pre-trained on the CheXpert [90] dataset as an MLC task to learn 14 common observations since the data scale of IU X-Ray is too

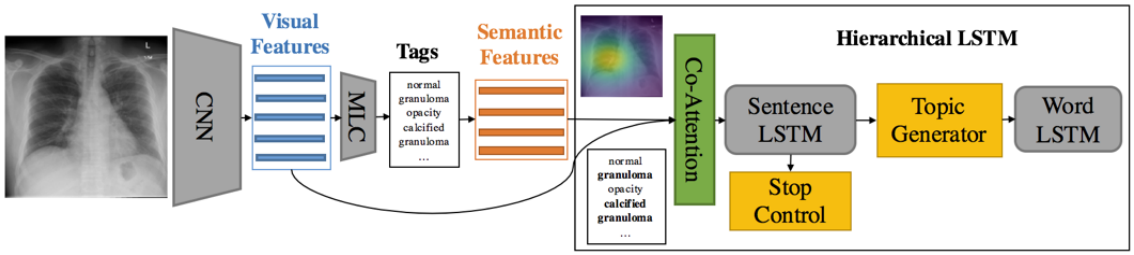


Figure 4.11: Illustration of the model proposed by Jing *et al.* The visual features are extracted from the last convolutional layer of the CNN (VGG-19 [25]) while the last two fully connected layers are used for the MLC task. Each tag is represented by a word-embedding vector that is used as a semantic feature vector. Both visual and semantic features are concatenated and fed into a co-attention model to obtain a joint context vector. The context vector is the input to a sentence LSTM, which produces a topic vector. Given a topic vector, the word LSTM generates the final sentence. Extracted from [86].

limited to obtain a robust performance. Then, the visual features from frontal and lateral views of the same patient are fed as a single input, using different fusion schemes. A sentence-level attention mechanism is employed to enforce the consistency between views. Finally, the most frequent medical concepts were extracted from the reports using SemRep [91], a tool to extract semantic knowledge representations from a biomedical text, and the encoder was fine-tuned to recognize such concepts, using a word-level attention mechanism. An overview of the proposed architecture can be seen in figure 4.12.

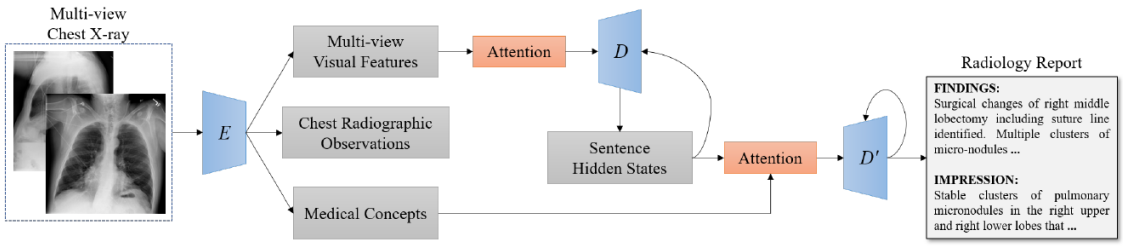


Figure 4.12: Overview of the architecture proposed by Yuan *et al.* E, D, and D' denote the encoder, sentence-level decoder, and word-level decoder, respectively. The sentence-level decoder is fed with visual features extracted from the encoder, and generates sentence hidden states. These are fed to the word-level decoder, which decodes the hidden states into natural language, generating the reports sentence-by-sentence. Extracted from [89].

Wang *et al.* [92] used an approach similar to Jing *et al.* [86] but with a flat LSTM decoder, instead of an hierarchical LSTM. Although they used a different chest x-ray dataset, results were significantly worse, presumably due to the flat nature of the decoder.

Lastly, Gale *et al.* [93] claimed that existing methods were failing to produce sufficient evidence to support a medical decision. This work focused on classifying hip fractures in pelvic

x-rays. They generated template-based reports and used simplified and standardized ground truth reports, explaining the extremely high results.

Although the aforementioned methods are not directly comparable, a table (4.1) with the scores of some of the methods can be seen below.

Table 4.1: Evaluation of biomedical image captioning methods with BLEU-1,2,3,4 (B1, B2, B3, B4), METEOR (MET), ROUGE-L (ROU), and CIDEr (CID) percentage scores. Adapted from [70].

Method	Dataset	B1	B2	B3	B4	MET	ROU	CID
Shin *et al.* [83]	IU X-Ray	79.3	9.1	0.0	0.0	—	—	—
Zhang *et al.* [84]	BCIDR	91.2	82.9	75.0	67.7	39.6	70.1	2.04
Jing *et al.* [86]	IU X-Ray	51.7	38.6	30.6	24.7	21.7	44.7	32.7
Yuan *et al.* [89]	IU X-Ray	52.9	37.2	31.5	25.5	34.3	45.3	—
Wang *et al.* [92]	Chest X-Ray	28.6	15.9	10.3	7.3	10.7	22.6	—
Gale *et al.* [93]	Frontal Pelvic X-Rays	91.9	83.8	76.1	67.7	—	—	—

4.5 Datasets

Despite the growing importance of medical image captioning, obtaining large-scale medical image datasets remains a challenge. To the best of our knowledge, there are only four publicly available datasets [94]. Indiana University Chest X-ray Collection (IU X-Ray), a dataset containing x-ray images; the Pathology Education Informational Resource (PEIR) digital library, a dataset of clinical photographs; ImageCLEF, a dataset with a combination of the previous two types of images; and the Bladder Cancer Image and Diagnostic Report dataset (BCIDR), a collection of whole-slide bladder tissue images. These datasets consist of medical images and associated texts. The text can be either a single sentence describing the image (PEIR Gross and ImageCLEF) or a medical report (IU X-Ray and BCIDR).

These datasets have a few drawbacks: IU X-Ray and PEIR Gross are rather small while ImageCLEF, the largest among them, has a large diversity of images. These three also share a significant class imbalance and have an insufficient number of disease-related terms, since they cover a wide range of diseases.

4.5.1 IU X-Ray

IU X-Ray [95] was made public through the Open Access Biomedical Image Search Engine (OpenI) [96]. This dataset consists of 7,470 chest x-rays and 3,955 reports written by radiologists. Each image-report pair is also coupled with a set of tags involving medical terms. The reports are structured into four sections: the "comparison" section contains previous information about the patient, the "indication" section contains the symptoms or reasons of examination, the "findings"

section describes the radiology observations, and the "impression" section depicts the final diagnosis. Only the "findings" section translates the visual content of the image, being appropriate for image captioning.

4.5.2 PEIR Gross

The PEIR digital library [97] provides another public access image dataset with 7,442 teaching images of gross lesions from 21 pathology sub-categories, coupled with associated captions.

4.5.3 ImageCLEF

The ImageCLEF dataset was first released in 2017 under the ImageCLEF caption task [98]. The dataset consists of around 184,000 figure-caption pairs extracted from PubMed Central (PMC) biomedical articles. Initially, the dataset introduced a lot of noise, with the presence of non-clinical images. Recently, the dataset has been reduced to radiology images.

4.5.4 BCIDR

The BCIDR dataset [85] contains 5,000 image-text pairs. A set of whole-slide images were taken from bladder tissue extracted from patients with the potential of developing bladder cancer. From this set of slide images, 1,000 images were randomly chosen. For each image, a pathologist provided a paragraph-like report, with variable length, describing the disease state. Then, four additional reports were written by doctors with no expertise on bladder cancer, with a total of five ground-truth reports per image. This feature offers a great advantage since many evaluation metrics achieve a higher correlation with human judgment when more reference sentences are provided [99]. Additionally, each image was classified as either normal, low-grade carcinoma, high-grade carcinoma, or insufficient information.

The overall summary of public access medical image captioning datasets can be seen below (table 4.2).

Table 4.2: Medical image captioning publicly available datasets. Adapted from [70].

Dataset	Images	Tags	Texts
IU X-Ray	7,470 x-rays	MESH & MTI extracted terms	3,955 reports
PEIR Gross	7,443 teaching images	top TF-IDF caption words	7,443 sentences
ImageCLEF	232,305 medical images	UMLS concepts	232,305 sentences
BCIDR	1,000 medical images	–	5,000 reports

Chapter 5

Implemented Architectures

This Chapter outlines the proposed approach to automatically generate medical imaging reports. Three different architectures were assessed using the IU X-Ray dataset: one was inspired by the original implementation of Xu *et al.* [66]. As previously mentioned in section 4.2.1, these authors were the first to introduce an attention-based mechanism for image captioning ($CNN + Att + LSTM$) that can be seen as an evolution of the baseline model proposed by Vinyals *et al.* ($CNN + LSTM$) [88]. The third approach was inspired by the work of Jing *et al.* [86], usually referred to as Co-Attention ($CNN + Att + HierLSTM$).

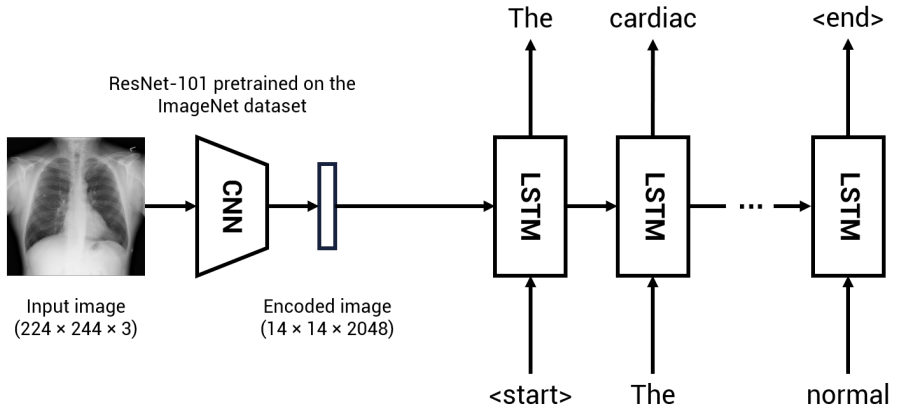
5.1 Implementation Details

In general, the aforementioned approaches [1] are based on an encoder-decoder architecture, thus coupling recent advances in computer vision and machine translation to generate natural language descriptions from images. CNNs are widely adopted for image-related tasks and used to encode images, while LSTMs, which have shown state-of-the-art performance on sequence tasks, are used to generate the textual descriptions. Xu *et al.* [66] and Jing *et al.* [86] also employ attention modeling to dynamically focus on specific parts of the input. The main difference between these works is the proposal of a hierarchical LSTM to cope with long paragraph generation by Jing *et al.* [86]. These concepts will be further detailed in the upcoming sections. A block diagram of each architecture can be seen in figure 5.1.

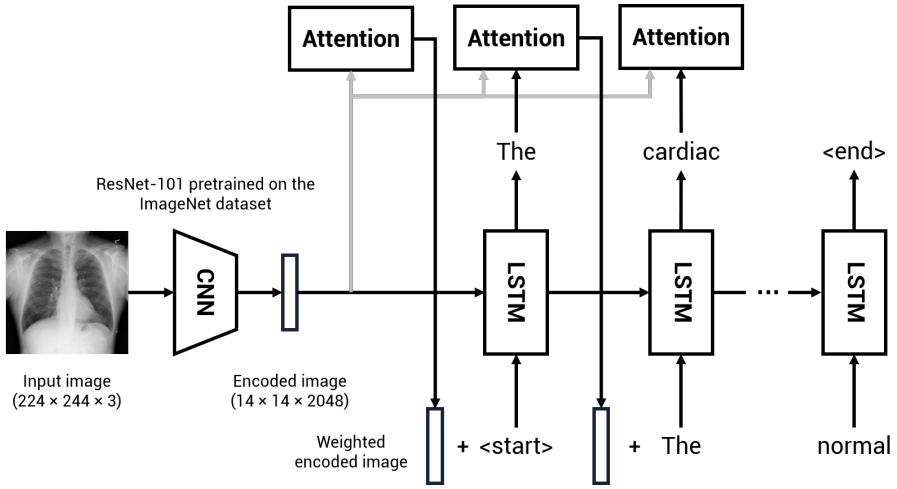
5.1.1 Encoder

The encoder stage involves feeding the images to a CNN that extracts feature vectors referred to as annotation vectors. Therefore, the CNN acts as a feature extractor, compressing the original image into smaller feature vectors while retaining the most important information, thus being called the encoder. These features are extracted from the last convolutional layer and are then used as input to the decoder stage.

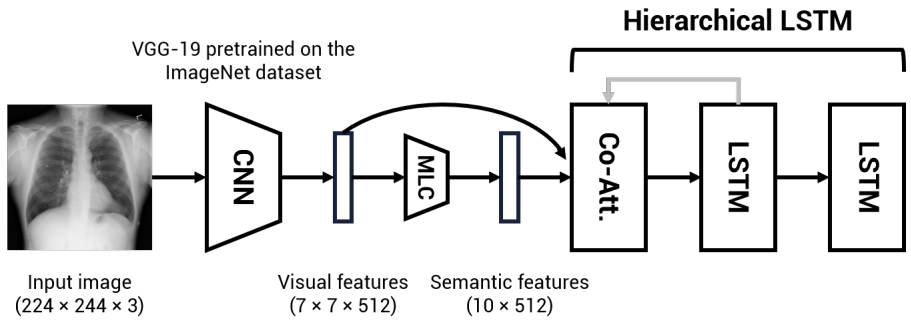
[1] the first two were adapted from an open-source project available on GitHub [100]



(a) CNN + LSTM



(b) CNN + Att. + LSTM



(c) CNN + Att. + Hier. LSTM

Figure 5.1: Block diagram of each architecture.

The CNN used in the first two approaches is the ResNet-101 [26]. In this work, the last two layers (pooling and linear layers) were removed, meaning only the convolutional module was

used to encode the image. Therefore, the encoder received an image with size $(224, 224, 3)$ and produced a feature map with size $(14, 14, 2048)$. The feature map was then flattened to $(196, 2048)$ and fed to the decoder that operated on this fixed-length representation.

The last approach replicates the original work proposed by Jing *et al.* [86], using a VGG-19 [25] as a visual feature extractor to produce an encoding with size $(14, 14, 512)$. Additionally, these features were fed into an MLC network to predict relevant tags for each image. To simplify, the fully connected layers of the network were used to perform this task. The word embeddings of the ten most likely tags for each image were used as semantic features.

5.1.2 Attention

Attention can be seen as an interface between the encoder and the decoder. The attention network implemented by Xu *et al.* [66] is a "soft" deterministic attention mechanism. It consists of a multi-layer perceptron conditioned by the encoder's output, previously referred to as annotation vectors, and the previous hidden state of the decoder, in each time-step. This mechanism generates weights for each annotation vector giving more or less importance to different image regions.

While the first approach does not have attention, the second one followed this mechanism by allowing the model to learn where to fix its gaze within an image while generating each word of the output caption (figure 5.2). This way, it is possible to gain insights into "where" and "what" the model focuses on to generate each word.

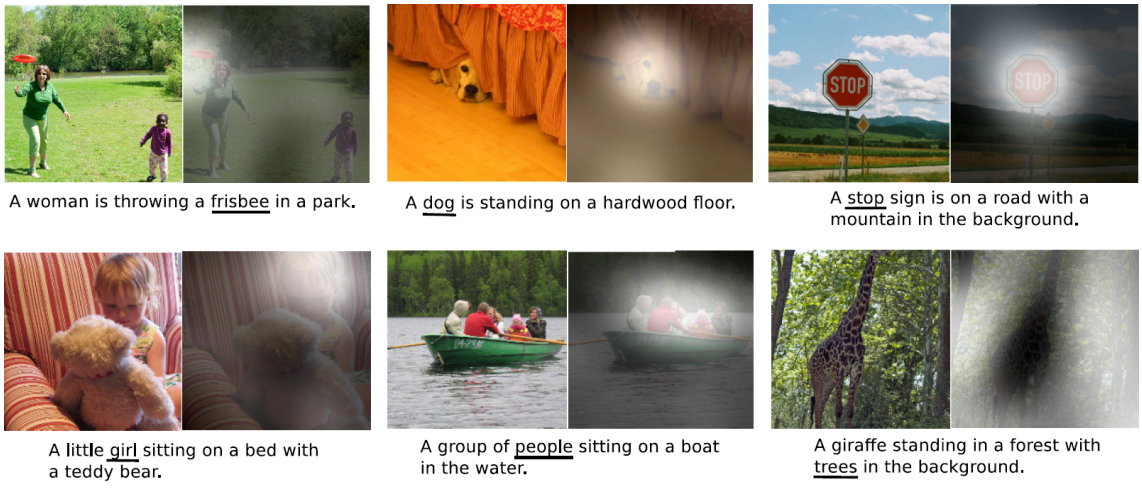


Figure 5.2: Visualization of the attention mechanism. White regions indicate the relevant regions and the underlined text indicates the corresponding word. Extracted from [66].

Jing *et al.* [86] state that visual attention alone is insufficient. They propose a co-attention mechanism that simultaneously attends to visual and semantic information. The co-attention network used in the third approach takes the visual and semantic features as input and produces a joint context vector capturing both visual and semantic attentions.

5.1.3 Decoder

An LSTM network is used as a decoder to generate the caption word by word. In the work proposed by Vinyals *et al.* [88] and described herein, the image was fed only once, at $t = -1$, to initialize the hidden and cell states. In subsequent steps, the input was the word embedding of each word, $W_e S_t$, of the caption, $S = (S_0, ..., S_N)$. Note that S_0 and S_N are the special START and END tokens, respectively. The authors established that feeding the image in each time step produces worse results, as the model overfits more easily. On the other hand, in Xu *et al.* [66], the previously generated word and the attention weighted annotation vectors are fed to the decoder in each time step. An illustrative example of such mechanism can be seen in figure 5.3.

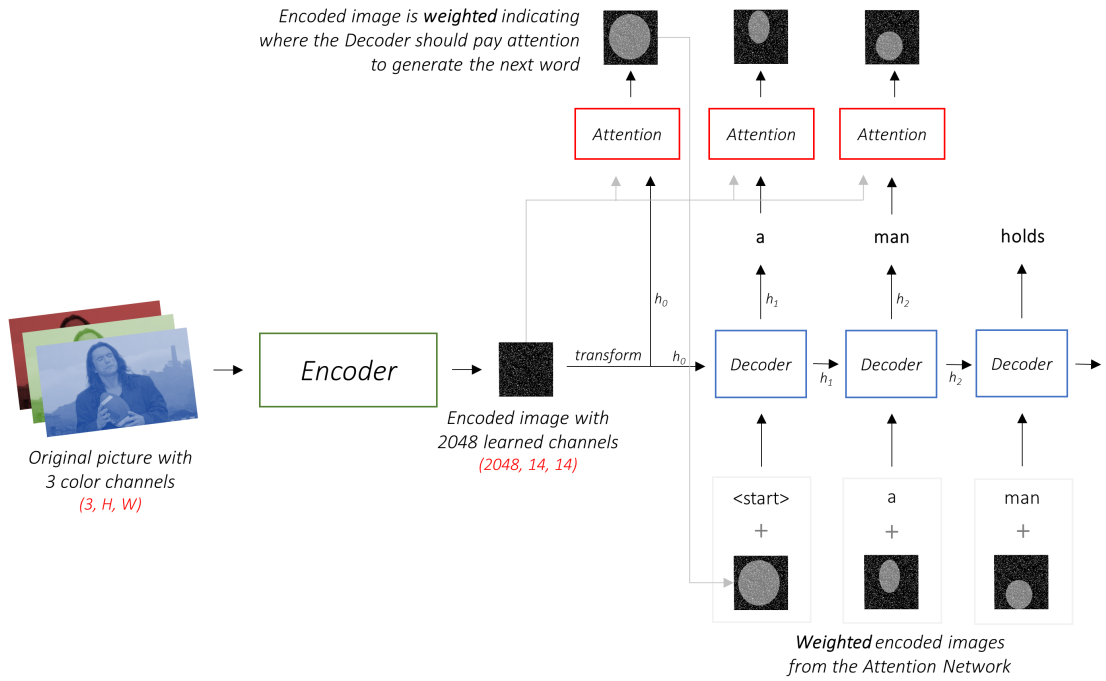


Figure 5.3: Example of an encoder-decoder architecture with an attention mechanism. The encoded image is initially fed to the decoder. In each time step, the decoder receives the weighted encoded image from the attention network. Extracted from [100].

Jing *et al.* [86] proposed a hierarchical LSTM that involves stacking two LSTMs together. In this third approach, the sentence LSTM took the joint context vector and produced high-level topic vectors and determined when to stop generating captions. It unrolled for S steps and combined the context vector and the current hidden state to create a topic vector. Concerning the stop control, the network took the previous and the current hidden states and produced a distribution over $\{STOP = 1, CONTINUE = 0\}$. If the probability to stop is higher than a predefined threshold, the word LSTM stops producing words.

The word-level LSTM received the topic vector and used it to initialize the hidden and cell states. The training procedure involved Teacher Forcing, i.e. feeding the real targets as the sub-

sequent inputs. While this method induces the model to converge faster, it may exhibit instability during inference. This way, the LSTM is conditioned by the number of sentences per caption and the number of words per sentence, meaning it only produces the exact number of words in the target caption, regardless of hitting the special END token earlier. The final reports are simply the concatenation of the generated sentences.

Note that to produce fixed-size Tensors from the captions in a batch, which are naturally of varying length, these were filled to the length of the largest sequence with PAD tokens. To avoid unnecessary computations, i.e., to avoid computing the loss over the padded regions, the captions were sorted by decreasing lengths and flattened by time-step, as can be seen in figure 5.4.

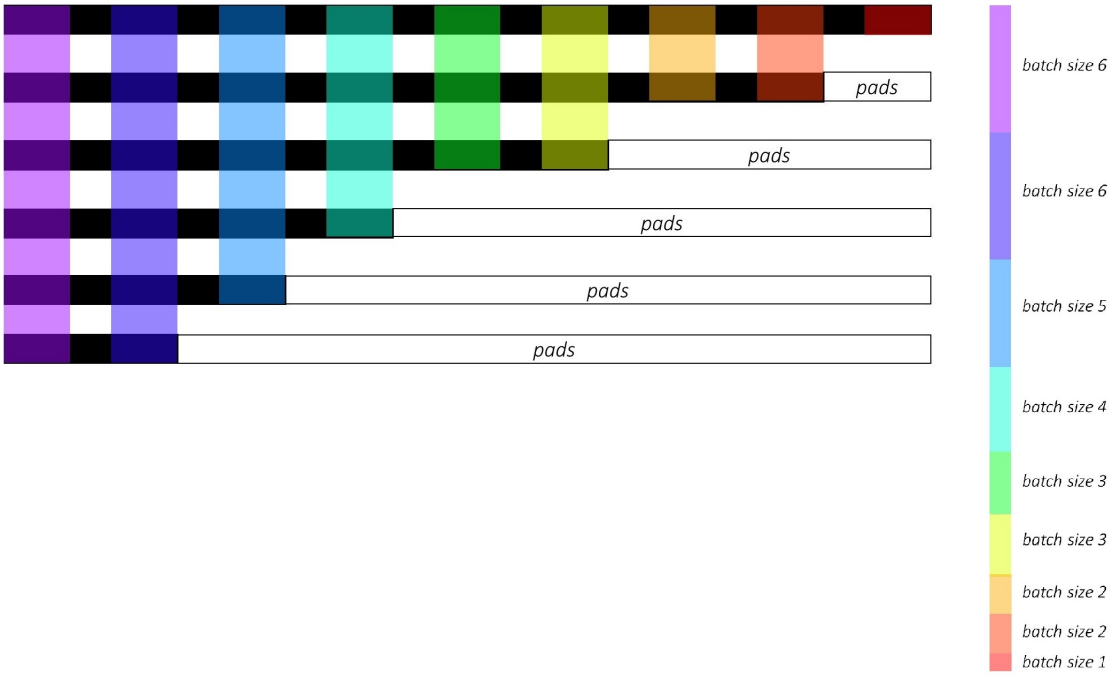


Figure 5.4: Padded sequences sorted by decreasing lengths (left) and packed sequence with the effective batch size in each time step (right). Adapted from [100].

5.2 Loss Functions

Concerning the paragraph generation, in the first two approaches the training loss is a cross-entropy loss over word distributions. The criterion receives as input the target word and a raw score for each word in the dictionary. The loss can be described as:

$$loss(x, word) = -\log\left(\frac{\exp(x[word])}{\sum_j \exp(x[j])}\right) = -x[word] + \log\left(\sum_j \exp(x[j])\right) \tag{5.1}$$

Concerning the attention weights, Xu *et al.* [66] propose a doubly stochastic regularization to encourage the model to pay equal attention to different regions of the image by forcing the

weights, α, at a single pixel, p, to sum to 1 across all time-steps, T. Otherwise, the model can neglect parts of the image. The regularization term can be described as:

$$\sum_{t}^{T} \alpha_{p,t} \approx 1 \tag{5.2}$$

As already stated, Jing *et al.* [86] introduce a multi-task learning framework with multiple loss functions, which was used in the third approach. The first step is an MLC task for tag prediction. The loss of this step is a binary cross-entropy loss, l_{tag}, between the probability distribution over all tags and a multi-hot encoded vector hinting at the presence or absence of each tag.

Next, the sentence LSTM produces S topic vectors and determines when to stop generating sentences. The word LSTM receives the topic vectors and generates the report. Concerning these steps, the loss is the combination of the cross-entropy losses over stop distributions, l_{sent}, and word distributions, l_{word}.

The overall training loss can be defined as:

$$\begin{aligned} l = {} & \lambda_{tag} l_{tag} \\ & + \lambda_{sent} \sum_{s=1}^{S} l_{sent}(p^{s}_{stop}, I\{s = S\}) \\ & + \lambda_{word} \sum_{s=1}^{S} \sum_{t=1}^{T_s} l_{word}(p_{s,t}, w_{s,t}) \end{aligned} \tag{5.3}$$

If the scales are too different, it is critical to set different weights, λ, to prevent one task from dominating the overall loss.

5.3 Training

An approach to deal with limited data in a given domain is to leverage data from another domain by performing transfer learning. This practice consists of reusing a trained network (ideally, a state-of-the-art network available in public repositories) by applying it to a different problem.

The CNN adopted in this work was the ResNet-101 [26] available in PyTorch's torchvision module and pre-trained on the 1000-class ImageNet dataset [20]. First, in all approaches, the weights of the pre-trained network were used, rather than training the model from scratch. Additionally, a different training procedure was implemented. The procedure consisted of fine-tuning the network using another dataset similar to the IU X-Ray, updating all of the model's parameters.

All networks were trained using the Adam [11] optimizer. In the first two approaches, the learning rates for the CNN (if fine-tuning) and the LSTM were $1e^{-4}$ and $4e^{-4}$, respectively.

5.4 Inference

In inference conditions, the model no longer uses Teacher Forcing. Instead, the LSTM is fed the previously generated word at each time-step. A common approach is using Beam Search to find the optimum sequence, as suggested by Vinyals *et al.* [88]. The "greedy" approach would be to choose the word with the highest score in each time-step. However, this technique may not sample the best sequence. When using Beam Search, the model considers k words in each decoding step and chooses the highest overall score among k candidate sequences.

5.5 Data

The results were obtained on the IU X-Ray dataset [95]. As stated in section 4.5, this dataset is one of the few publicly available datasets for medical image captioning. The IU X-Ray dataset consists of a collection of radiology examinations from the Indiana University, comprising images and reports by radiologists, and it can be accessed through the OpenI search engine [96]. Before being released to the public, the dataset contained 8,121 images and 3,996 reports. However, some had to be removed for privacy reasons, leaving a subset of 7,470 images of frontal- and lateral-view chest x-rays and 3,955 reports, since more than one image can be associated with the same report.

Each report is divided into four sections: the "comparison" section carries background information about the patient, such as the medical history; the "indication" section holds the symptoms or reasons of examination; the "findings" section describes the radiology observations; the "impression" section draws the final diagnosis. For this purpose, only the "findings" section was used as input to generate image descriptions. However, a few captions may also contain information that was not inferred from the image content.

All image-report pairs are also associated with a set of tags. This encoding was performed using the Medical Subject Heading (MeSH) [101, 102] indexing. The most frequently assigned tags are shown in table 5.1.

Among the 3,955 reports, 28 were blank (excluding 40 image-caption pairs) and 101 were not linked to any image, leaving a total of 3,826 reports and 7,430 images. There were also 489 reports with no findings (approximately 13% of the remaining reports) plus six reports missing the impression, but none missing both. However, reports without findings seemed to have the corresponding information misplaced in the "impression" section. To avoid losing data, the latter was taken into account whenever the report was missing the "findings" section.

The data set was then split into training, validation, and test by randomly selecting 500 images for validation and 500 images for testing. Words with a frequency lower than five were stored as an unknown token (UNK). Finally, only captions with a maximum length of 100 were sampled.

Concerning the images, the pre-trained encoder expected them to be resized to 256×256 and loaded in to a range of $[0,1]$, before being normalized using $mean = [0.485, 0.456, 0.406]$ and $dev = [0.229, 0.224, 0.225]$.

Table 5.1: The 10 most frequently assigned codes among the original 3,996 reports, excluding the 1,526 'normal' coded reports. Extracted from [95].

Code	Source	# Coded reports (% of 2470 non-normal reports)
Cardiomegaly	MeSH	375 (15.1)
Pulmonary atelectasis	MeSH	347 (14.0)
Calcified granuloma	MeSH/RadLex	284 (11.5)
Aorta/tortuous	MeSH/RadLex	253 (10.2)
Lung/hypoinflated	MeSH/RadLex	245 (9.9)
Opacity/lung base	RadLex	203 (8.2)
Pleural effusion	MeSH	172 (6.9)
Lung/hyperinflation	MeSH/RadLex	164 (6.6)
Cicatrix/lung	MeSH	148 (5.9)
Calcinosis/lung	MeSH	141 (5.7)

The dataset used to perform fine-tuning on the encoder is publicly available on Kaggle [103]. It consists of 5,856 chest x-ray images divided into two classes: normal or pneumonia. The class division was conducted by physicians before being cleared for training AI systems. It is important to mention that images classified as "pneumonia" account for 73% of the dataset, which exhibits a great class imbalance.

5.6 Evaluation Metrics

One of the most widely adopted evaluation metrics in medical image captioning is BLEU [71]. By default, the BLEU score calculates the cumulative 4-gram BLEU score (BLEU-4). In this case, the weight that is given to each 1-, 2-, 3-, and 4-gram scores is 25%. It is common to report results using cumulative BLEU-1 to BLEU-4 scores. ROUGE-L and CIDEr, alternative metrics that have been shown to correlate better with human judgment, are also provided for comparison. These were calculated using a standard evaluation tool [104].

Chapter 6

Results and Discussion

This chapter discusses the results obtained from the three models described previously, which are reported using BLEU, the most commonly adopted metric, ROUGE-L, and CIDEr. The results obtained from additional experiments (different attention mechanisms and encoders) are also provided in subsequent sections.

6.1 Architecture Comparison

6.1.1 Quantitative Results

The validation loss and the BLEU score typically show a negative correlation. However, the validation loss and the BLEU score showed a break of correlation in the later stages of training, with the loss function starting to degrade before the BLEU score did. That said, the training procedure used early stopping on the validation loss as a regularization technique. Table 6.1 provides a summary of the quantitative performance of the models on the IU X-Ray. Results from the original papers are also provided in Table 6.2.

Table 6.1: Results obtained from the three implemented approaches on the IU X-Ray dataset.

Method	Pre-trained Encoder	B1	B2	B3	B4	ROU	CID
CNN + LSTM [88]	ImageNet	18.3	10.6	7.4	5.6	23.3	33.2
	Chest X-Ray (Kaggle)	19.0	10.9	7.5	5.7	23.2	31.9
CNN + Att. + LSTM [66]	ImageNet	35.4	17.4	10.8	7.3	23.3	34.1
	Chest X-Ray (Kaggle)	**35.6**	**17.7**	**11.0**	**7.6**	**23.4**	**36.3**
CNN + Att. + Hier. LSTM [86]	ImageNet	29.0	14.6	7.8	4.5	23.1	34.2
	Chest X-Ray (Kaggle)	23.8	12.8	7.2	4.1	22.5	33.3

It is relevant to notice that the BLEU score (and other evaluation metrics initially designed for machine translation or image captioning) is not truly appropriate for the report generation task. BLEU, in particular, can be misleading. Models can achieve high scores while suffering from missing or repeated information [105].

Table 6.2: Results extracted from the original papers. (—) indicates an unknown metric.

Method	Dataset	B1	B2	B3	B4	ROU	CID
CNN + LSTM [88]	Flickr8k	63.0	41.0	27.0	—	—	—
	Flickr30k	66.3	42.3	27.7	18.3	—	—
	MS COCO	66.6	46.1	32.9	24.6	—	—
CNN + Att. + LSTM [66]	Flickr8k	67.0	44.8	29.9	19.5	—	—
	Flickr30k	66.7	43.4	28.8	19.1	—	—
	MS COCO	70.7	49.2	34.4	24.3	—	—
CNN + Att. + Hier. LSTM [86]	IU X-Ray	51.7	38.6	30.6	24.7	44.7	32.7

6.1.2 Qualitative Results

Table 6.3 shows some of the generated reports from the second model using a beam size of 5. It is visible that the model is not capable of identifying rare abnormalities within an image. The first example exhibits a chest X-Ray of a healthy patient without any findings. Here, the model generates a report that preserves the meaning even if it does not perfectly reproduce the reference report. The second example introduces a minor irregularity, while the third example shows an X-Ray with numerous abnormalities. In both situations, the model fails to identify these anomalies.

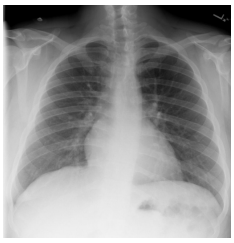
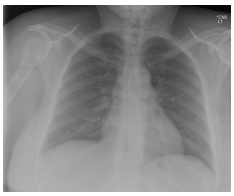
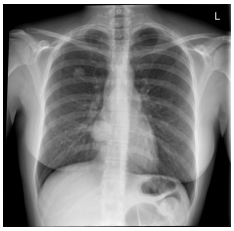
The visualization of the attention learned by the model also adds an extra layer of interpretability to its output, as shown by Xu *et al.* [66]. However, the attention mechanism is not always capable of linking the findings precisely with the corresponding words. An example of the visualization of the attention mechanism can be seen in figure 6.1.

6.1.3 Discussion

The first challenge presented in Chapter 1 pertains to the shortage of large-scale medical datasets with reports. Hence, the IU X-Ray dataset was used since it is one of the few publicly available datasets. However, this dataset alone offers numerous obstacles. First, there is a considerable class imbalance: "normal" cases account for 38% of the dataset [83] while the remaining data cover a wide range of conditions, which leads to rare occurrences of disease-related terms [70]. Furthermore, the Health Insurance Portability and Accountability Act of 1996 (HIPAA) forces the reports to go through a data anonymization process to remove any information that can identify an individual, thus leaving some captions illegible.

Concerning the task of generating reports itself, there are also some drawbacks. Traditional image captioning methods typically do not handle medical datasets but shorter and less elaborate pieces of text. A medical imaging report is a heterogeneous source of information — it consists of multiple sentences focusing on various aspects of the image, and it can provide more or less detail depending on the writing style of the doctor. As a consequence, there is higher variability in the data content and length. Furthermore, given the potentially disastrous consequences of a failed

Table 6.3: Example of reports generated by the second approach. The sentences in bold are examples of abnormalities, while the "xxxx" represents obliterated information.

Chest X-Ray	**Ground Truth**	**Generated Report**
	Normal heart size and mediastinal contours. The lungs are clear. There is no pneumothorax or pleural effusion. The xxxx are unremarkable.	The heart is normal in size. The mediastinum is unremarkable. The lungs are clear.
	The heart is normal size. The mediastinum is unremarkable. There is no pleural effusion, pneumothorax, or focal airspace disease. **Mild degenerative changes are present within the spine.**	The lungs are clear bilaterally. Specifically, no evidence of focal consolidation, pneumothorax, or pleural effusion. Cardio mediastinal silhouette is unremarkable. Visualized osseous structures of the thorax are without acute abnormality.
	There are **2 xxxx masses within the right chest**, largest over the right heart xxxx measuring up to 3.8 x 3.4 cm. The appearance is concerning for **metastatic disease**, given the history of right-sided breast cancer.	The heart and lungs have xxxx xxxx in the interval. Both lungs are clear and expanded. Heart and mediastinum normal.

diagnosis, reports must be not only readable but also clinically accurate. To do so, the model must have a correct understanding of the content of an image.

State-of-the-art methods on medical imaging report generation are still unable to diversify language and find abnormal regions within an image [106]. The first approach was the simplest among the three models, which resulted in worse performance. Although the model was capable of generating coherent and grammatically correct reports, it showed a bias towards healthy individuals, meaning it could not identify anomalies within an image. The second model was the only one capable of generating more diversified reports although it was also not always able to accurately match the correct findings to an image. The third model presented a behavior similar to the first model, also exhibiting a substantial amount of repeated information. Overall, the methods seem to focus more on language information and heavily memorize captions from the training dataset rather than focusing on visual information. Goyal *et al.* [107] point out that while this can result in high superficial performance, the model does not have a proper understanding of the visual content. This phenomenon can also explain why there is not considerable variation in terms of BLEU score between experiments using an encoder previously trained on the ImageNet or fine-tuned on the Kaggle dataset.

Finally, this study shows there is a compromise between the model's complexity and the amount and quality of the available training data. When there is a lack of data, depending on its

Figure 6.1: Visualization of the attention mechanism from the second approach on the IU X-Ray dataset. White regions indicate the relevant regions.

complexity, the model will either underfit or overfit the training data, with both situations resulting in poor performance. Vinyals *et al.* and Xu *et al.* validated their models on larger datasets such as the MS COCO [108] that contains over 300.000 images with five captions each. As previously

mentioned, providing several reference sentences is also helpful in achieving a better agreement with human judgment.

6.2 Impact of individual components

Given the second model, which showed the most promising performance, the impact of using different CNNs as the encoder was studied. The following table (6.4) shows the performance of the second model using the ResNet-101, DenseNet-121, and GoogleNet as the encoder.

Table 6.4: Results obtained from the second model using different CNNs as the encoder.

Method	Encoder	B1	B2	B3	B4	ROU	CID
CNN + Att. + LSTM [66]	ResNet-101	35.6	17.7	11.0	7.6	23.4	**36.3**
	DenseNet-121	35.5	17.9	**11.3**	**7.9**	**24.2**	30.7
	GoogleNet	**36.3**	**18.0**	11.2	7.7	24.1	28.5

6.3 Attention mechanism

An additional experiment consisted of adding the MLC block of Jing *et al*'s model [86] to the second one (CNN + Att. + LSTM) [66] to introduce semantic attention. Similarly to the visual attention network, the semantic attention is updated at each decoding step with respect to the hidden state and fed as input to the LSTM alongside the word embeddings and the visual attention.

Table 6.5 provides a summary of the quantitative performance of the model on the IU X-Ray, using different CNNs as the encoder.

Table 6.5: Results obtained from ours co-attention approach on the IU X-Ray dataset.

Method	Encoder	B1	B2	B3	B4	ROU	CID
Semantic Attention	ResNet-101	18.3	10.5	7.3	5.6	23.3	33.1
	DenseNet-121	**19.2**	**11.0**	**7.6**	**5.8**	**23.4**	**34.4**
	GoogleNet	18.3	10.5	7.3	5.6	23.3	33.1
Co-Attention	ResNet-101	19.4	11.1	7.7	5.8	23.3	**34.4**
	DenseNet-121	35.3	17.6	10.8	7.3	23.8	25.1
	GoogleNet	**36.3**	**18.0**	**11.2**	**7.7**	**24.1**	28.5

The results show that semantic attention alone is not enough, with the performance decreasing considerably. These results validate the ones obtained by Jing *et al.* [86] that had already shown that semantic attention alone does not seem to help on report generation. Adding semantic attention to visual attention did not have a significant impact on the overall performance of the model

as well. In fact, the results deteriorated when using the ResNet-101, possibly due to the size of the embedding layer of the MLC block being higher in this case. Although a word embedding with a small dimensionality is usually not expressive enough, a very large dimensionality can lead to over-fitting and tends to increase model complexity [109].

Chapter 7

Conclusion and Future Work

Explaining what triggered a model's decision is one of the most challenging and relevant problems nowadays. In Healthcare, in particular, it is crucial to build models that are understandable to humans to increase user acceptance. However, there is little or no research on generating textual explanations in an unsupervised fashion. Also, there are still a few challenges concerning the generation of detailed and precise descriptions from medical images.

Healthcare providers usually write reports describing their judgments. Teaching a model to mimic this behavior is a means of providing a layer of interpretability. In that sense, this work involved developing a model that could accurately describe pieces from an image that would ultimately result in an accurate diagnosis. However, all models, in general, missed vital visual information, which led to poorly diversified reports and incorrect diagnoses. In light of these results, the second model [66] is the most promising for further development. Nevertheless, testing with different encoders and attention mechanisms suggested that the model is heavily relying on the language information rather than focusing on the visual content.

In summary, three different approaches were assessed. The most effective model was the one that had the best balance between complexity and the available data, i.e., the second model. Moreover, it allows for further development to enhance the quality of the reports. Ultimately, the performance of a model will always be intrinsically subject to the amount and quality of the available data.

A potential research path is to replace the recurrent architecture with a self-attention mechanism, i.e., replace the decoder in the encoder-decoder architecture with the decoder layer of the transformer. Recently, there have been efforts to introduce the transformer model [60] to the image captioning task. Initially designed for machine translation tasks, the transformer achieves state-of-the-art performance and requires significantly less training time. These authors also showed it generalizes well to other sequence problems. In this sense, future work could follow the same approach as Zhu *et al.* [110]. The authors proposed using a CNN to modify the image to the same dimensions of the word embeddings. Then, the second sub-layer of the transformer takes the encoding as input. The transformer involves a stack of attention and feed-forward modules, thus being able to train in one forward pass similarly to the CNN.

Another possible research direction is to use pre-trained word embeddings rather than training them from scratch. As previously mentioned, using pre-trained embeddings is useful when there is a large volume of rare words, which is observed in the IU X-Ray dataset. It also leads to faster training and usually helps performance. This task requires using a state-of-the-art method for effectively learning dense vector representations (usually pre-trained on a general domain corpus) and a sufficiently large amount of text data in a medical domain. According to Mikolov *et al.* [46], Skip-gram models, in particular, work well with small amounts of data and representing rare words, thus being suitable for medical datasets that are usually smaller. An alternative is using Bidirectional Encoder Representations from Transformers for Biomedical Text Mining (BioBERT) [111], a language representation model that is already pre-trained on a biomedical domain corpora (PubMed abstracts and PMC full-text articles).

In the long term, this work could be extended to produce textual explanations without direct supervision. The model could be pre-trained on self-supervised tasks in the text domain, such as learning how to predict masked words from sentence context, with image classification as an auxiliary task to ensure that the explanations are consistent with the image content. However, this research direction introduces additional challenges due to its highly subjective nature and the difficulty of determining what a proper explanation is.

References

[1] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73:1–15, 2018.

[2] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[3] Types of neural networks (and what each one does!) explained. https://towardsdatascience.com/types-of-neural-network-and-what-each-one-does-explained-d9b4c0ed63a1. (Accessed on 01/11/2020).

[4] Why deep learning over traditional machine learning? https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063. (Accessed on 01/11/2020).

[5] Weight initialization in neural networks: A journey from the basics to kaiming. https://towardsdatascience.com/weight-initialization-in-neural-networks-a-journey-from-the-basics-to-kaiming-954fb9b47c79. (Accessed on 01/11/2020).

[6] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[8] Activation functions with derivative and python code: Sigmoid vs tanh vs relu. https://medium.com/@omkar.nallagoni/activation-functions-with-derivative-and-python-code-sigmoid-vs-tanh-vs-relu-44d23915c1f4. (Accessed on 01/12/2020).

[9] How dead neurons hurt deep learning training ? | by joel chao | joelthchao | medium. https://medium.com/joelthchao/how-dead-neurons-hurt-training-5fc127d8db6a. (Accessed on 07/23/2020).

[10] Léon Bottou. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8):12, 1991.

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul):2121–2159, 2011.

[13] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[14] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[15] Applied deep learning - part 4: Convolutional neural networks. https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2. (Accessed on 01/12/2020).

[16] Transfer learning explained - the integrate.ai blog - medium. https://medium.com/the-official-integrate-ai-blog/transfer-learning-explained-7d275c1e34e2. (Accessed on 02/11/2020).

[17] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[18] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. 01 2017.

[19] Waseem Rawat and Zenghui Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[21] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[24] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[25] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[27] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[28] Cnn architectures — lenet, alexnet, vgg, googlenet and resnet. https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848. (Accessed on 01/11/2020).

[29] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[30] Interpretability vs. accuracy: The friction that defines deep learning. https://towardsdatascience.com/interpretability-vs-accuracy-the-friction-that-defines-deep-learning-dae16c84db5c. (Accessed on 01/16/2020).

[31] Been Kim and Finale Doshi-Velez. Interpretable machine learning: the fuss, the concrete and the questions. *ICML Tutorial on interpretable machine learning*, 2017.

[32] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91, 2018.

[33] Navigating the sea of explainability - towards data science. https://towardsdatascience.com/navigating-the-sea-of-explainability-649672aa7bdd. (Accessed on 01/15/2020).

[34] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[35] explainable ai (xai) research must go beyond the four walls of research labs. https://towardsdatascience.com/explainable-artificial-intelligence-xai-research-must-go-beyond-the-four-walls-of-research-labs-16faacbf9dff. (Accessed on 01/16/2020).

[36] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). *arXiv preprint arXiv:1711.11279*, 2017.

[37] Jacek M Zurada, Aleksander Malinowski, and Ian Cloete. Sensitivity analysis for minimization of input data dimension for feedforward neural network. In *Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94*, volume 6, pages 447–450. IEEE, 1994.

[38] AH Sung. Ranking importance of input parameters of neural networks. *Expert systems with Applications*, 15(3-4):405–411, 1998.

[39] Javed Khan, Jun S Wei, Markus Ringner, Lao H Saal, Marc Ladanyi, Frank Westermann, Frank Berthold, Manfred Schwab, Cristina R Antonescu, Carsten Peterson, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine*, 7(6):673–679, 2001.

[40] Stephen Bazen and Xavier Joutard. The taylor decomposition: A unified generalization of the oaxaca method to nonlinear models. 2013.

[41] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015.

[42] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[43] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[44] Gregory Plumb, Denali Molitor, and Ameet S Talwalkar. Model agnostic supervised local explanations. In *Advances in Neural Information Processing Systems*, pages 2515–2524, 2018.

[45] A comprehensive guide to natural language generation. https://medium.com/sciforce/a-comprehensive-guide-to-natural-language-generation-dd63a4b6e548. (Accessed on 02/06/2020).

[46] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[47] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[48] John G Kemeny and J Laurie Snell. *Markov chains*. Springer-Verlag, New York, 1976.

[49] Machine learning — hidden markov model (hmm) - jonathan hui - medium. https://medium.com/@jonathan_hui/machine-learning-hidden-markov-model-hmm-31660d217a61. (Accessed on 02/06/2020).

[50] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[51] MI Jordan. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986. Technical report, California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science, 1986.

[52] Rnns to write like shakespeare - branav kumar gnanamoorthy - medium. https://medium.com/@gnabr/rnns-to-write-like-shakespeare-226609863cd1. (Accessed on 01/26/2020).

[53] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[54] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[55] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE, 2011.

[56] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[57] Illustrated guide to lstm's and gru's: A step by step explanation. https://towardsdatascience.com/illustrated-guide-to-lstms-and-grus-a-step-by-step-explanation-44e9eb85bf21. (Accessed on 02/06/2020).

[58] Understanding lstm networks – colah's blog. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. (Accessed on 05/22/2020).

[59] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[61] 10.1. attention mechanisms — dive into deep learning 0.8.0 documentation. http://d2l.ai/chapter_attention-mechanisms/attention.html. (Accessed on 06/15/2020).

[62] Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. Multimodal neural language models. In *International Conference on Machine Learning*, pages 595–603, 2014.

[63] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CSUR)*, 51(6):1–36, 2019.

[64] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.

[65] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. Guiding the long-short term memory model for image caption generation. In *Proceedings of the IEEE international conference on computer vision*, pages 2407–2415, 2015.

[66] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[67] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[68] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Incorporating copying mechanism in image captioning for learning novel objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6580–6588, 2017.

[69] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. Stylenet: Generating attractive visual captions with styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3137–3146, 2017.

[70] Vasiliki Kougia, John Pavlopoulos, and Ion Androutsopoulos. A survey on biomedical image captioning. *arXiv preprint arXiv:1905.13302*, 2019.

[71] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[72] Chin-Yew Lin and FJ Och. Looking for a few good metrics: Rouge and its evaluation. In *Ntcir Workshop*, 2004.

[73] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[74] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013.

[75] Desmond Elliott and Frank Keller. Image description using visual dependency representations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1292–1302, 2013.

[76] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluation the role of bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

[77] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[78] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[79] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[80] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881, 2017.

[81] Thomas Schlegl, Sebastian M Waldstein, Wolf-Dieter Vogl, Ursula Schmidt-Erfurth, and Georg Langs. Predicting semantic descriptions from medical images with convolutional neural networks. In *International Conference on Information Processing in Medical Imaging*, pages 437–448. Springer, 2015.

[82] Pavel Kisilev, Eli Sason, Ella Barkan, and Sharbell Hashoul. Medical image captioning: Learning to describe medical image findings using multi-task-loss cnn. *Deep Learning for Precision Medicine, Riva del Garda, Italy*, 2016.

[83] Hoo-Chang Shin, Kirk Roberts, Le Lu, Dina Demner-Fushman, Jianhua Yao, and Ronald M Summers. Learning to read chest x-rays: Recurrent neural cascade model for automated image annotation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2497–2506, 2016.

[84] Zizhao Zhang, Yuanpu Xie, Fuyong Xing, Mason McGough, and Lin Yang. Mdnet: A semantically and visually interpretable medical image diagnosis network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6428–6436, 2017.

[85] Zizhao Zhang, Pingjun Chen, Manish Sapkota, and Lin Yang. Tandemnet: Distilling knowledge from medical images using diagnostic reports as optional semantic references. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 320–328. Springer, 2017.

[86] Baoyu Jing, Pengtao Xie, and Eric Xing. On the automatic generation of medical imaging reports. *arXiv preprint arXiv:1711.08195*, 2017.

[87] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A hierarchical approach for generating descriptive image paragraphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 317–325, 2017.

[88] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[89] Jianbo Yuan, Haofu Liao, Rui Luo, and Jiebo Luo. Automatic radiology report generation based on multi-view image fusion and medical concept enrichment. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 721–729. Springer, 2019.

[90] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 590–597, 2019.

[91] Semantic knowledge representation. https://semrep.nlm.nih.gov/. (Accessed on 04/21/2020).

[92] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, and Ronald M Summers. Tienet: Text-image embedding network for common thorax disease classification and reporting in chest x-rays. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9049–9058, 2018.

[93] William Gale, Luke Oakden-Rayner, Gustavo Carneiro, Andrew P Bradley, and Lyle J Palmer. Producing radiologist-quality reports for interpretable artificial intelligence. *arXiv preprint arXiv:1806.00340*, 2018.

[94] Imane Allaouzi, M Ben Ahmed, B Benamrou, and M Ouardouz. Automatic caption generation for medical images. In *Proceedings of the 3rd International Conference on Smart City Applications*, pages 1–6, 2018.

[95] Dina Demner-Fushman, Marc D Kohli, Marc B Rosenman, Sonya E Shooshan, Laritza Rodriguez, Sameer Antani, George R Thoma, and Clement J McDonald. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 23(2):304–310, 2016.

[96] https://openi.nlm.nih.gov. https://openi.nlm.nih.gov/. (Accessed on 02/07/2020).

[97] Peir digital library. http://peir.path.uab.edu/library/. (Accessed on 02/07/2020).

[98] Carsten Eickhoff, Immanuel Schwall, Alba García Seco de Herrera, and Henning Müller. Overview of imageclefcaption 2017-image caption prediction and concept detection for biomedical images. In *CLEF (Working Notes)*, 2017.

[99] Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014.

[100] sgrvinod/a-pytorch-tutorial-to-image-captioning: Show, attend, and tell | a pytorch tutorial to image captioning. https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning. (Accessed on 02/07/2020).

[101] Frank B Rogers. Medical subject headings. *Bulletin of the Medical Library Association*, 51(1):114–116, 1963.

[102] Medical subject headings - home page. https://www.nlm.nih.gov/mesh/meshhome.html. (Accessed on 07/07/2020).

[103] Chest x-ray images (pneumonia) | kaggle. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia. (Accessed on 06/04/2020).

[104] salaniz/pycocoevalcap: Python 3 support for the ms coco caption evaluation tools. https://github.com/salaniz/pycocoevalcap. (Accessed on 06/25/2020).

[105] Inlg2018.key. https://inlg2018.uvt.nl/wp-content/uploads/2018/11/INLG2018-YoavGoldberg.pdf. (Accessed on 06/22/2020).

[106] Yuan Li, Xiaodan Liang, Zhiting Hu, and Eric P Xing. Hybrid retrieval-generation reinforced agent for medical image report generation. In *Advances in neural information processing systems*, pages 1530–1540, 2018.

[107] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

[108] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[109] Zi Yin and Yuanyuan Shen. On the dimensionality of word embedding. In *Advances in Neural Information Processing Systems*, pages 887–898, 2018.

[110] Xinxin Zhu, Lixiang Li, Jing Liu, Haipeng Peng, and Xinxin Niu. Captioning transformer with stacked attention modules. *Applied Sciences*, 8(5):739, 2018.

[111] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.