Radiology Report Summarization with Large Language Models

by

Mahmoud Idlbi

A thesis submitted to the
Graduate Program in Computing
in conformity with the requirements for
the degree of Master of Science

Queen's University
Kingston, Ontario, Canada
May 2025

Abstract

Cancer remains a leading cause of mortality worldwide, with timely and accurate assessments of disease progression critical for optimal treatment. In routine oncology practice, repeated radiology examinations yield numerous reports—each containing detailed *Findings* and a more concise *Impression*—that clinicians must synthesize to make informed treatment decisions. Motivated by the time-intensive and error-prone nature of manually reviewing these longitudinal imaging histories, this thesis proposes a two-stage fine-tuning strategy for radiology report summarization using both decoder-only and encoder-decoder large language models (LLMs), specifically LLaMa 3.1-8B-Instruct and T5-Base.

In the first stage, we fine-tune each model on MIMIC-CXR, a large publicly available dataset of chest radiographs with free-text radiology reports, to transform the *Findings* into an *Impression*. Due to its large parameter count, the LLaMa model was fine-tuned using QLoRA—a quantized adaptation approach that enables memory-efficient training—while T5-Base was fine-tuned using standard LoRA. Our results demonstrate substantial performance gains over the respective base models. For LLaMa, improvements of 174.18%, 115.57%, 129.48%, 16.08%, and 111.15% were observed in BLEU-4, ROUGE-L, METEOR, BERTScore, and F1-RadGraph, respectively. Similarly, the fine-tuned T5-Base model achieved BLEU-4, ROUGE-L, and

F1-RadGraph scores of 31.25, 47.57, and 40.73—representing increases of 677.36%, 255.18%, and 337.31% over the base model.

In the second stage, we refine both models on a small, custom-curated dataset of 174 *Impressions* and their associated summaries. The objective of this stage is to enable the model to summarize individual *Impressions* while emphasizing longitudinal and clinically relevant changes. This setup facilitates iterative summarization of radiology reports over time. The two-stage fine-tuned LLaMa model significantly outperforms its base version on this task, achieving improvements of 456.88%, 209.15%, 57.01%, 15.19%, and 35.65% across BLEU-4, ROUGE-L, METEOR, BERTScore, and F1-RadGraph, respectively. The two-stage T5 model achieved improvements of 35.16%, 20.09%, 4.39%, 3.05%, and 7.96% across BLEU-4, ROUGE-L, METEOR, BERTScore, and F1-RadGraph, respectively. While both models performed well overall, the two-stage LLaMa model outperformed the T5 model on the downstream *Impression* summarization task, whereas T5 achieved the best results on the MIMIC-CXR task. These findings suggest that LLaMa's decoder-only architecture may offer advantages in the two-stage fine-tuning strategy.

Acknowledgments

I would like to express my deepest gratitude to the Queen's University community—its faculty, staff, and students. The spirit of curiosity and dedication within this community has made my time at Queen's a pleasure.

I extend my sincere appreciation to my supervisor, Dr. Amber Simpson, for her invaluable guidance, encouragement, and support throughout this thesis. Her insights and feedback have been invaluable in refining my work and pushing me to explore new directions.

I am also grateful to all members of the Simpson Lab, whose camaraderie and support have made this journey both productive and rewarding. In particular, I would like to thank Dr. Mohammad Hamghalam and Jianwei Yue for their generosity in sharing their knowledge and taking the time to support my research and academic pursuits at Queen's University.

A special thank you to Dr. Richard Do of the Memorial Sloan Kettering Cancer Center for his contribution in curating a custom dataset used in this work, as well as for his valuable clinical insights that helped shape the development of the algorithm.

Finally, I want to express my deepest gratitude to Zoé Lord, for her love, encouragement, and companionship. And to my family, thank you for always celebrating my victories and supporting me throughout my journey.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Cancer remains one of the leading causes of mortality worldwide, and effective treatment relies heavily on timely, accurate identification of disease progression and response to therapy [77]. Patients with cancer undergo many imaging studies over the course of their treatment.

Each image has a corresponding report generated by a radiologist. These reports capture findings from modalities such as computed tomography (CT), magnetic resonance imaging (MRI), and X-ray and are essential documents for guiding treatment decisions and monitoring therapeutic efficacy [7, 51]. The primary components of a radiology report include the technical description of the imaging study, detailed *Findings*, and a final *Impression* section that synthesizes the results [7].

The *Findings* section provides a comprehensive description of what was observed during the scan. It documents any abnormalities that may indicate disease progression, metastasis, or other clinically significant changes. The *Findings* are typically organized by anatomical region, listing any pathological observations in the organs or systems examined. These include descriptions of lesions, masses, lymph node involvement, or changes in comparison to prior imaging studies. For example, as shown

in Figure 1.1, the *Findings* can indicate the presence or absence of metastatic disease in specific organs.

The *Impression* section provides the most important findings in a concise clinical summary. It is arguably the most significant section of the report, as it provides a quick, clear interpretation of the imaging results to inform treatment decisions [22]. In oncology, particularly in cases of metastatic disease, the *Impression* summarizes the presence or absence of metastases, any new developments, and how these findings compare to previous scans. This section allows oncologists and referring physicians to immediately focus on the most critical information without the need to parse the more extensive technical details contained in the *Findings* section. As seen in Figure 1.1, the *Impression* concisely provides key findings, such as metastases in the lungs, liver, and spleen, providing a clear summary of the patient's disease status.

Recent advances in natural language processing (NLP) and large language models (LLMs) have shown great potential for processing medical text, as they have already demonstrated capabilities in tasks such as clinical text summarization and named entity recognition [94, 36]. However, most existing methods focus on summarizing a single radiology report's *Findings* into an *Impression* [53, 10, 18, 59, 94]. This is likely due to the lack of publicly available datasets for longitudinal radiology report summarization. Summarizing a list of *Impressions* would capture disease evolution over time and enable radiologists to quickly review key changes across serial imaging reports. This could improve clinical workflows by directing radiologists to the most significant reports in a patient's history, providing concise summaries for referring physicians and nurses, and alleviating cognitive workload on medical staff.

Motivated by this critical gap, the present work proposes a two-stage fine-tuning

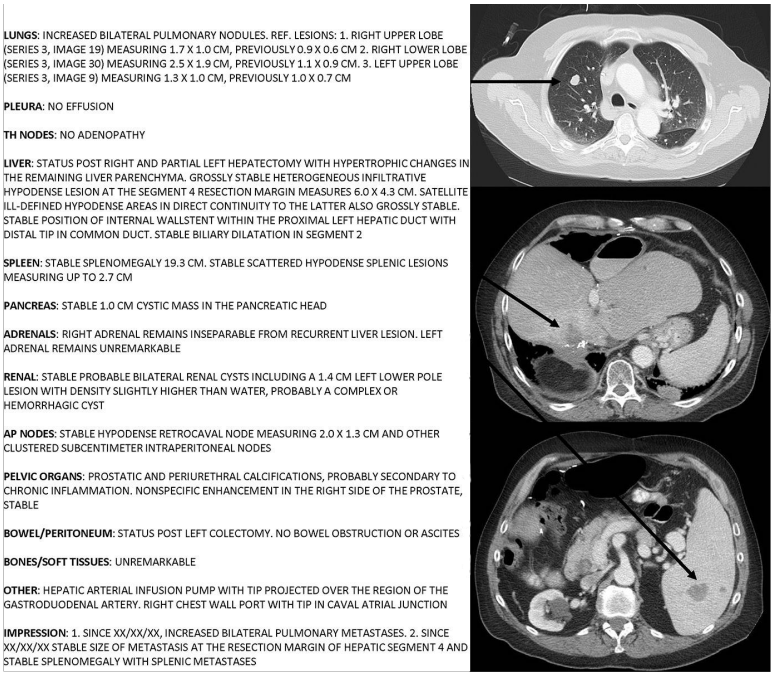

Figure 1.1: An example of a structured radiology report detailing the Findings and summarizing the presence of metastatic disease in the lungs, liver, and spleen, as observed in a 74-year-old man with colorectal cancer and a history of hepatic resection [23].

approach to automatically detect key time points in a patient's imaging history and to generate concise, clinically valuable summaries that emphasize metastatic disease trends. We first adapt an LLM to the specialized language of radiology by learning to transform *Findings* into an *Impression*, and then refine the model to produce shorter, longitudinal summaries of *Impressions*. As there are no publicly available datasets of *Impressions* and their corresponding summary, we curated our own dataset of 174 *impressions* and their corresponding summary. The task of summarizing a list of *Impressions* can be looked at as a task of summarizing a single *Impression* iteratively for the full list. Due to our small dataset size, we take this approach and our presented results are based on single *Impression* summarization. We aim to reduce

the cognitive load on clinicians and ensure that important therapeutic opportunities are not overlooked.

1.1 Problem Statement

Patients with metastatic disease undergo repeated imaging studies over extended time periods, generating a substantial volume of radiology reports. Depending on disease progression, treatment phase, and clinical needs, cancer patients may undergo between 10 to 30 imaging studies annually, with metastatic cases often requiring even more frequent monitoring [71]. Over time, this results in a rapidly growing collection of reports that clinicians must manually review to track tumor burden, detect new lesions, and assess treatment response. This process is both time-intensive and prone to error, particularly as clinicians manage an expanding caseload alongside the complexity of longitudinal disease assessment [7, 78]. The writing style of different radiologists and institutions also varies, complicating longitudinal patient history analysis and cross-institutional studies. As a result, critical information, such as disease progression, emerging metastases, or significant changes in tumor response, can be overlooked or inconsistently reported. To address this gap, there is a clear need for an automated system capable of summarizing key findings across multiple radiology reports, focusing on temporal trends and clinically significant events. The goal of this work is to develop and evaluate fine-tuned large language models—specifically LLaMA 3.1 8B-Instruct [48] and T5 [68]—for longitudinal summarization and clinical reasoning tasks, producing concise, structured summaries of longitudinal radiology data. Both models were trained and assessed on domain-specific radiology datasets to compare their effectiveness in capturing clinically relevant longitudinal insights.

1.2 Objective

The primary objective of this thesis is to develop and evaluate an automated framework for summarizing radiology reports using a two-stage fine-tuning approach, applied to both the LLaMA 3.1 8B-Instruct and T5 models. In the first stage, each model is trained on the MIMIC-CXR dataset, transforming the *Findings* section of each report into a concise *Impression*. This stage serves as an initial test of each model's capacity to learn radiology-specific language patterns and produce clinically relevant summaries. The resulting performance is compared against previously published models that tackle similar radiology summarization tasks.

In the second stage, both models are further fine-tuned on a smaller, custom-curated dataset composed of 174 *Impressions* collected from 20 patients, each with longitudinal radiology histories. This step is designed to assess whether the models, having acquired foundational knowledge of radiology reporting in the first stage, can extend that capability to a more specialized task of summarizing *Impressions* with a focus on temporal trends. The performance in this second stage offers insight into the models' ability to generalize and adapt to related tasks when only a limited dataset is available—serving as a proof of concept for broader clinical applications of automated radiology summarization.

1.3 Hypothesis

This thesis hypothesizes that relatively small language models—specifically LLaMA 3.1 8B-Instruct and T5—can be effectively fine-tuned to summarize radiology reports and produce *Impressions* that align with those written by radiologists. We further hypothesize that a two-stage fine-tuning strategy will enable each model to

learn radiology-specific language and summarization skills from a large public dataset (MIMIC-CXR), and subsequently apply that knowledge to a much smaller, custom dataset (174 *Impressions* and their corresponding summaries). We predict that this multi-stage approach will allow both models to generate outputs that closely match expert clinical summaries, despite the limited size of the second-stage dataset.

1.4 Contributions

In this thesis, we demonstrate that the LLaMA 3.1 8B-Instruct model, despite having fewer parameters than the larger alternatives in its family (LLaMA 3.1-70B and LLaMA 3.1-405B), can be fine-tuned to summarize radiology *Findings* into concise, clinically relevant *Impressions* with relatively competitive performance. To this end, we compare the performance of our model to other published approaches.

A key methodological contribution of this work is the introduction of a two-stage fine-tuning strategy that extends the LoRA framework across both phases. In Stage 1, models are adapted to radiology-specific language using the large-scale MIMIC-CXR dataset. In Stage 2, the same models are further fine-tuned on a novel, custom-curated dataset consisting of 174 longitudinal radiology cases, each containing an *Impression* with an expert-written summary. This two-phase setup enables effective knowledge transfer from general to institution-specific reporting styles and showcases how LoRA-based adaptation can remain efficient and effective even under limited data conditions. While demonstrated in the context of radiology, this approach is broadly applicable to other domains where summarization from structured or semi-structured medical text is required, such as pathology, cardiology, or operative reports.

In addition to evaluating LLaMA, we extend this investigation to T5, an encoder-decoder model architecture, to compare how differently structured models handle the same two-stage fine-tuning process. Our comparative results offer insight into the generalizability, specialization, and trade-offs between decoder-only and encoder-decoder language models in clinical summarization tasks.

1.5 Organization of Thesis

This thesis is structured into five chapters. Chapter 2 provides a review of related work, covering fundamental concepts in artificial intelligence, NLP, and summarization techniques, with a particular focus on transfer learning and fine-tuning LLMs for radiology report summarization. It also discusses evaluation metrics for assessing LLM-generated summaries. Chapter 3 details the methodology, including model selection, dataset description, fine-tuning procedures, summarization processes, and evaluation frameworks. Chapter 4 presents the results, reporting on the performance of fine-tuned models on the tasks of MIMIC-CXR and impression summarization, along with a comparison to existing work. Chapter 5 concludes the thesis with a discussion of findings, limitations, future directions, and implications for medical practice.

Chapter 2

Related Work

2.1 Artificial Intelligence

Artificial Intelligence (AI) broadly encompasses computational methods designed to imitate or augment human intelligence, allowing machines to perceive their environment, make decisions, and solve problems. Early conceptual work on AI can be traced back to the 1950s, when researchers such as McCarthy and Turing explored the possibility of machines that exhibit human-like intelligence through symbolic logic and reasoning [56, 84]. Over time, AI has evolved into a rich and interdisciplinary field encompassing computer science, cognitive science, neuroscience, and engineering [74]. Today, AI techniques drive innovations across various domains, from voice-activated personal assistants and recommendation systems to self-driving vehicles and medical imaging analysis. In each of these applications, the core methodologies revolve around machine learning, an approach that enables systems to learn patterns from data, adapt to new examples, and refine decision-making strategies.

Central to modern AI is the optimization of cost or reward functions. Most algorithms aim to minimize some form of prediction error or maximize a reward signal

while learning representations from large datasets [82]. In supervised learning, models adjust their internal parameters to minimize the discrepancy between predicted outputs and known, labeled data; this process can be achieved through gradient descent, wherein iterative refinements steer the system toward more accurate predictions. Alternatively, in reinforcement learning, agents interact with an environment and optimize a policy to maximize cumulative rewards over time. Through trial-and-error, these agents become adept at tasks such as game-playing and robotic control. Many powerful AI models combine these learning paradigms, applying techniques that focus on both data-driven features and the structuring of reward-based objectives [29].

Thanks to advances in computational power and the abundance of digital data, AI has rapidly progressed over the last few decades. High-capacity neural architectures, such as deep convolutional and recurrent networks, have made it feasible to handle large and complex datasets [43, 32]. These architectures often benefit from extensive pre-training, which captures generalizable patterns before fine-tuning models for specific downstream tasks, a process similar to how humans acquire fundamental skills before mastering specialized domains [22]. As a result, AI systems today can reliably interpret images, parse language, identify complex temporal trends, and even generate convincing text and visual artifacts [29].

While the field of AI deals with a broad spectrum of cognitive and perceptual abilities, it is the specialized subdomain of **Natural Language Processing (NLP)** that concentrates on understanding and generating human language. The next sections discuss how NLP methodologies have evolved from rule-based systems to sophisticated neural networks.

2.2 Natural Language Processing

Language functions as a structured system of symbols and rules that are used to convey information between individuals. **NLP** has emerged as tool to allow computers to understand and process human language. It has utilities in many applications including machine translation, email spam detection, sentiment analysis, information extraction, and text summarization. Recently, NLP has bridged the communication gap between users and machines, allowing interactions in everyday human language without requiring technical expertise [39]. NLP is an interdisciplinary field that combines insights from artificial intelligence, linguistics, psychology, and cognitive science [38]. It plays a crucial role in modern technology, allowing machines to interpret, analyze and generate language-based input and output.

At its core, NLP is divided into two primary components: Natural Language Understanding (NLU) and Natural Language Generation (NLG). **NLU** focuses on extracting meaning from language through various linguistic levels, including phonology (the study of sound), morphology (word structure), syntax (sentence formation), semantics (meaning of words and phrases) and pragmatics (context-dependent meaning). For example, phonology examines how sounds are arranged to form meaningful utterances, while syntax ensures sentences are grammatically correct. NLU systems are used in applications such as chatbots and sentiment analysis, where machines must understand language input [39]. On the other hand, **NLG** involves producing linguistically correct text based on internal data representations, turning structured information into sentences or paragraphs to convey information.

The origins of NLP lie in the development of rule-based systems that relied heavily on structured grammars, such as **Context-Free Grammars** (CFGs). These systems

enabled syntactic parsing, where words were grouped into phrases and phrases into sentences to analyze language [14]. In the 1950s, researchers began to explore machine translation and created foundational rule-based algorithms [3, 54, 96]. However, progress was slow due to the inherent ambiguity and complexity of human language [4], which led researchers to spend the following decades refining the theoretical foundations of the field. During the 1980s and 1990s, the focus in NLP shifted from rule-based systems to statistical methods [55]. Researchers began developing probabilistic models that could better manage language variability and ambiguity by learning from large text corpora. These probabilistic models became widely adopted approaches for language tasks [65], including speech recognition [66], predicting the next word from a sequence [8], and cursive handwriting learning [27].

Around the same time, the groundwork for neural networks that would transform the field of artificial intelligence in the 2010s was being carried out. In 1986, Rumelhart et al. [72] published a paper introducing the *backpropagation learning method* in neural networks. Arguably one of the most significant advancements in neural networks, this method adjusts connection weights to minimize the error between the network's output and the desired output, allowing "hidden" units (intermediate layers) to capture key task features and patterns. For instance, in a feedforward network trained on language data, the first hidden layer might detect individual word patterns, the next layer might capture syntactic structures like noun phrases or verb-object pairs, and deeper layers could learn high-level relationships between concepts. These transformations happen progressively, allowing the network to learn complex, non-linear mappings from input to output. Figure 2.1 shows a typical feedforward neural network, where information flows in a single direction from input to output

through one or more hidden layers. While this architecture is useful for many tasks, it processes each input independently, without any mechanism to remember or relate it to prior inputs.

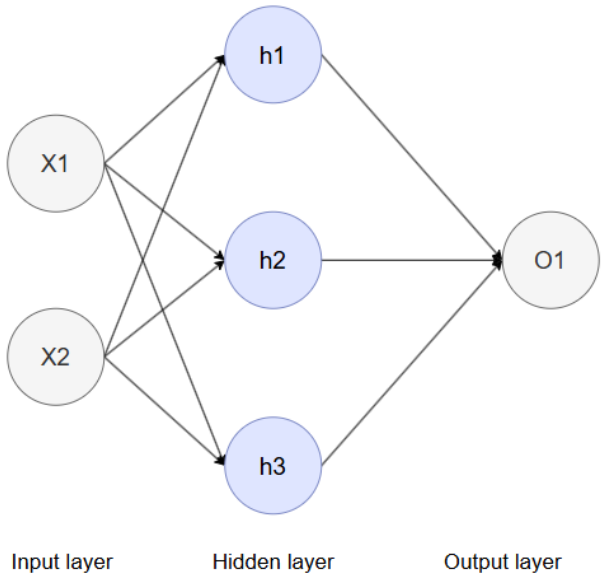

Figure 2.1: Feedforward neural network with one hidden layer.

Rumelhart's [72] introduction of hidden layers also made it possible to extend neural networks to sequential data through the development of **recurrent neural networks** (RNNs). RNNs incorporate feedback connections that allow the network to maintain a dynamic memory of previous inputs. This makes RNNs particularly well-suited for tasks where the order and timing of inputs matter—such as language modeling, time series forecasting, and speech recognition—because the meaning of a given input often depends on what came before it [75]. However, recurrent neural networks are prone to the problem of vanishing and exploding gradients, where gradients either diminish or grow excessively as they are backpropagated through many time steps [5]. This issue arises because repeated multiplication of gradients

over long sequences causes their values to become very small or very large, making it difficult for the network to learn dependencies over long sequences [63]. In the case of vanishing gradients, the network struggles to capture long-term dependencies, while exploding gradients can lead to unstable updates during training.

In 1997, Hochreiter and Schmidhuber introduced **Long Short-Term Memory** (LSTM) [32], a specialized recurrent neural network with a gradient-based learning algorithm that employs gating mechanisms to manage the information flow. The LSTM uses three key gates, input, forget and output gates, which control which information is added, retained, or discarded from the cell state. This gating structure helps preserve gradients across long sequences, allowing LSTMs to effectively capture both short- and long-term dependencies without suffering from vanishing or exploding gradients.

Although LSTMs are very powerful in capturing long-term dependencies, they still assume that the structures of input-to-output sequences are order-dependent. The development of **sequence-to-sequence** (seq2seq) models was a significant advancement in NLP, as it allowed the mapping of input sequences into output sequences with minimal assumptions about the structure of the sequences [12]. Sutskever et al. (2014) [81] introduced the seq2seq framework using LSTM networks composed of an **encoder** to map an input sentence into a fixed-length vector and a **decoder** to translate this vector back into a target sentence. In experiments on English-to-French translation, their model achieved a higher level of overlap between generated and target outputs as compared to phrase-based machine translation methods. Importantly, the model effectively produced representations of phrases and sentences that preserved word order and syntactic dependencies. However, relying on a fixed

length vector in the seq2seq approach presented a limitation when dealing with long or complex sentences. To address this bottleneck, several works [1, 52] introduced an attention mechanism that allowed the model to *focus* on different parts of the input sequence dynamically as it generated each target word. The soft attention mechanism works by assigning a weight to each word in the input sequence based on its relevance to the current word that is being generated in the target sequence. These weights are then used to create a weighted sum of the hidden states of the input, producing a unique context vector for each target word that dynamically emphasizes the most relevant parts of the input sentence.

Building on the success of attention, the introduction of self-attention in the **Transformer** model by Vaswani et al. (2017) [86] significantly advanced NLP by removing the need for recurrent and convolutional networks. The model follows an encoder-decoder structure in which both components use self-attention and position-wise feed-forward layers, with residual connections and layer normalization to support stable training, as shown in Figure 2.2. By processing the entire input sequence in parallel, Transformers address the bottlenecks of sequential processing found in RNNs, making them more efficient at handling long sequences. The original Transformer architecture is especially well-suited to sequence-to-sequence tasks such as machine translation, where an encoder captures the contextual representation of the input and a decoder then generates the output sequence one token at a time.

Subsequent variants of the Transformer architecture have evolved into three primary categories based on their use of encoder and decoder components. BERT is an example of an *Encoder-only* model, designed for understanding and representation-based tasks rather than sequence generation. It is trained using a masked language

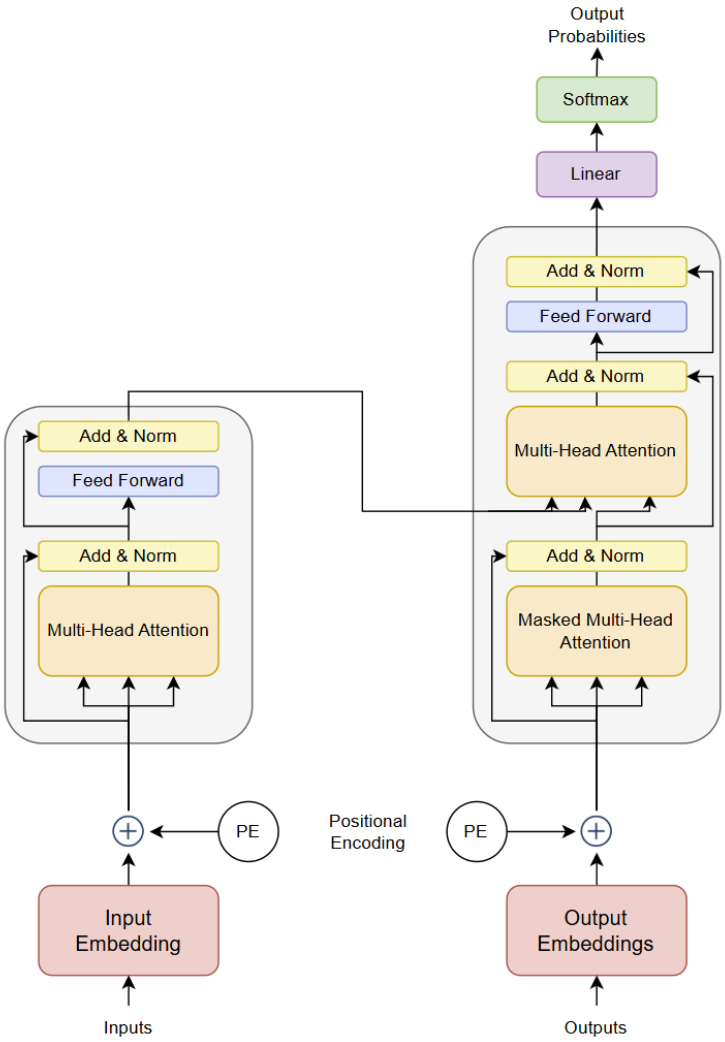

Figure 2.2: Transformer architecture consisting of an encoder (left) and decoder (right). Both use stacked self-attention and feed-forward layers with residual connections and layer normalization. The decoder includes an additional masked attention mechanism to prevent access to future tokens during training.

modeling (MLM) objective, where random words in the input are masked, and the model learns to predict these missing words based on the surrounding context. This pre-training approach enables BERT to develop deep contextual representations, making it highly effective for downstream tasks that require a rich understanding of input

text. Encoder-only architectures like BERT and its successors (e.g., RoBERTa [47], ALBERT [42]) are predominantly used in applications where bidirectional comprehension is essential, such as text classification, named entity recognition, and information retrieval.

Encoder-decoder models are central to high-performing sequence-to-sequence applications, including neural machine translation, abstractive summarization, and speech-to-text processing. Models in this category maintain a clear link between input and output sequences by using an encoder to process the input into a contextual representation, which the decoder then uses to generate the final output. T5 [68] is a well-known example that excels in tasks like translation and summarization. These encoder-decoder frameworks still form the backbone of many state-of-the-art translation systems and serve as a foundation for multimodal architectures that integrate text, images, and speech.

Models like GPT [67, 9] and LLaMa [48], on the other hand, are *decoder-only* architectures that focus on autoregressive text generation. They use causal self-attention, meaning each token only considers the tokens before it, thus preventing future tokens from leaking information during generation. Because these models do not require a separate encoding stage, they can directly produce text from a prompt, making them ideal for open-ended tasks such as conversational AI, code generation, and text writing.

2.3 Summarization using Natural Language Processing

Text summarization in natural language processing can be divided into extractive and abstractive methods. Extractive summarization selects key sentences or phrases

directly from the source text to preserve only essential information. Techniques such as heuristic rules and graph-based algorithms, such as TextRank [57], are commonly employed in this approach. However, extractive methods often result in summaries that can be disjointed or overly rigid [73, 10].

In contrast, abstractive summarization generates new sentences that convey the main ideas of the source material more naturally, similar to human-written summaries. This approach requires a deeper understanding of semantics and the ability to reformulate the text structure effectively; therefore, it requires more complex methods and larger datasets. Based on Bahdanau et al.'s [1] translation model, Rush et al. (2015) [73] used a neural attention-based model composed of a feedforward neural network combined with a textual input encoder and a beam-search decoder. This model is trained on a single-sentence summarization task and showed significant performance gains on existing benchmarks compared to previous work. Several works have also applied an attention-based neural model in extractive summarization tasks that allow summaries to avoid repetitions and include more key sentences by taking into account sentence-level or word-level contextual relations [11, 70, 28].

Multiple researchers also integrated the attention model with recurrent neural networks. In [15], Chopra et al. (2016) propose a conditional recurrent neural network (RNN) model that generates summaries by focusing on contextually important words through a convolutional attention-based encoder. In similar work, Nallapati et al. (2016) [58] improved the basic attentional encoder-decoder RNN architecture to improve abstractive summarization using a bidirectional **Gated Recurrent Unit** (GRU) for the encoder and a unidirectional GRU for the decoder. Their model also leverages a switching generator-pointer model to handle out-of-vocabulary words by

pointing to their source positions and a hierarchical attention mechanism that identifies relevant sentences and words within them.

Further improvements in the field of summarization would be achieved through reinforcement learning; where Paulus et al. (2017) [64] introduced a deep reinforced encoder-decoder based on the seq2seq model that achieved state-of-the-art results in CNN / Daily Mail and New York Times summarization tasks. Their algorithm leverages a hybrid reinforcement learning objective that minimizes a maximum-likelihood loss at each decoding step while also maximizing a discrete metric through policy learning. The model utilizes intra-temporal attention at each decoding step to focus on specific parts of the encoded input sequence, the decoder's hidden state, and the previously generated words; with access to this information, the model can avoid repetition in newly generated words. Several other works at the time, including Paulus' et al., would leverage copy mechanisms to handle rare and out-of-vocabulary words [64, 30, 76].

2.3.1 Summarization using LLMs

Current state of the art approaches in summarization, and NLP in general, are built using transformer-based models. Zhang et al. (2019) [91] introduced a novel approach that integrates BERT as a foundational component in a two-stage encoder-decoder architecture, specifically designed for abstractive summarization. The BERT-based encoder generates context-rich embeddings of the source text, capturing syntactic and semantic relationships between words. The two-stage decoder then refines the summary generated through a unique process. In the first stage, the model drafts an initial summary using a left-to-right transformer-based decoder. In the second stage,

it uses a bidirectional BERT-based mechanism to mask each word in the draft and refine it by predicting each masked token, effectively allowing the model to review and improve the summary based on global context. Similarly, Liu and Lapata (2019) [46] extended BERT for both extractive and abstractive summarization by developing BERTSUM, a framework that utilizes a document-level encoder and layers of inter-sentence transformers to capture relationships between sentences. This approach further enables effective extraction of key sentences and fine-tuning for abstractive tasks. Furthermore, BERT's bidirectional design allows these models to capture both the *preceding* and the *following* context during refinement.

Several works [44, 50] have leveraged denoising autoencoders to improve text generation. Lewis et al. (2020) [44] introduced BART, a model that combines BERT's bidirectional encoding with autoregressive decoding, pre-training on corrupted text using noise injection techniques such as token masking, deletion, and sentence shuffling. Meanwhile, Lou et al. (2022) [50] proposed a simplified convolutional denoising autoencoder (SCDAE) designed to reconstruct chaotic signals with minimal preprocessing, demonstrating its ability to recover structured patterns from noisy inputs. By learning to predict missing or distorted information, denoising techniques help models develop a stronger understanding of underlying patterns, making them more robust in generating coherent summaries even when mistakes are present in the source text.

Recent advances in large language models have shown that they can effectively perform complex NLP tasks, such as summarization, without explicit task-specific training. LLMs are pre-trained on massive corpora, allowing them to capture linguistic patterns, contextual cues, and knowledge representations. Brown et al. (2020)

[9] demonstrated this potential with GPT-3, a 175-billion-parameter autoregressive model trained to complete text based on input prompts. In their experiments, they found that GPT-3 could generate high-quality summaries in few-shot scenarios - where only a few examples are provided - despite not being explicitly trained for summarization. Kojima et al. (2024) [41] further explored the zero-shot reasoning capabilities of LLMs, demonstrating that these models could tackle complex multi-step reasoning tasks without task-specific examples by employing a simple prompt adjustment. Their approach, known as Zero-shot-CoT (Chain of Thought), involves adding the phrase "Let's think step by step" before each answer prompt, which significantly boosts performance on challenging reasoning benchmarks. Remarkably, this method led to substantial accuracy improvements in various tasks, such as arithmetic problem solving (e.g., MultiArith and GSM8K), symbolic reasoning, and logical inference, with gains from 17.7% to 78.7% in MultiArith and from 10.4% to 40.7% in GSM8K when using the InstructGPT model. [9, 41] findings reveal that the underlying capabilities of LLMs for broad cognitive tasks can often be activated by minimal prompt adjustments.

2.4 Transfer Learning and Fine-Tuning Large Language Models

2.4.1 Transfer Learning

Transfer learning is a machine learning paradigm where knowledge gained from one domain or task—the source domain—is leveraged to improve performance on a related but distinct task, known as the target domain. Unlike traditional approaches that assume training and testing data must share the same feature space and distribution, transfer learning enables models to adapt more efficiently by reusing learned feature

representations or parameters. This is particularly valuable when labeled data in the target domain are scarce [61].

A key motivation for transfer learning is its ability to mitigate data scarcity while enhancing computational efficiency. Many real-world applications lack sufficient labeled data, making it impractical to train models from scratch. Transfer learning addresses this issue by leveraging generalizable knowledge from models trained on large-scale, data-rich domains. In natural language processing, models such as BERT and GPT acquire broad linguistic understanding from extensive unlabeled corpora, enabling them to effectively transfer linguistic knowledge to new tasks with minimal additional training data [61, 9]. Similarly, in computer vision, convolutional neural networks such as ResNet are commonly pre-trained on vast labeled datasets such as ImageNet [19], allowing their learned features to be reused in specialized domains, including medical imaging or satellite imagery analysis.

Transfer learning is broadly categorized into three types based on the relationship between the source and target domains and tasks: inductive transfer learning, transductive transfer learning, and unsupervised transfer learning. These categories show how knowledge transfer can be applied effectively, addressing differences in feature spaces, distributions, or learning objectives.

Inductive Transfer Learning occurs when the tasks in the source and target domains are different, regardless of whether the domains are the same. This approach assumes that labeled data are available to train a predictive function in the target domain. The goal is to leverage knowledge from the source task to improve performance on the target task, often using shared representations or parameters. For example, a classifier trained on textual data from one domain (e.g., news articles) might be

adapted to a sentiment analysis task in another domain (e.g., product reviews). Inductive transfer learning is commonly used when there are limited labeled data in the target domain but enough to guide the adaptation process [61].

Transductive Transfer Learning applies when the source and target tasks are the same, but the domains differ. In this setting, the labeled data are abundant in the source domain, but there are no labeled data in the target domain. Transductive learning focuses on adapting models to the target domain using the structure or distribution of the unlabeled data of the target domain. A typical application is domain adaptation, such as adapting a sentiment classifier trained on English reviews to predict sentiments in Chinese reviews, where the features (e.g., languages) differ, but the task remains consistent. Transductive transfer learning often employs techniques such as feature alignment to bridge domain differences [61].

Unsupervised Transfer Learning addresses scenarios in which both the source and target domains have unlabeled data and different tasks [6]. This setting has been explored less but holds potential for tasks like clustering or dimensionality reduction in the target domain. For example, unsupervised feature learning methods in the source domain might help uncover better representations for clustering data in the target domain [25]. This type of transfer learning is beneficial in exploratory data analysis or when labeled data is unavailable in both domains [61].

2.4.2 Fine-Tuning Language Models

Fine-tuning is a specialized form of transfer learning that involves adapting a pre-trained model specifically by further training it on task-specific labeled data. Unlike

broader transfer learning approaches that may reuse only general features or representations, fine-tuning directly updates the parameters of an existing model to optimize performance on a given downstream task. One of the first works to demonstrate the effectiveness of fine-tuning in NLP was introduced by Howard and Ruder (2018) [34], labeled the Universal Language Model Fine-tuning for Text Classification (ULMFiT) methodology. It consists of three main stages: (1) pre-training a general domain language model (LM) on a large corpus to learn broad linguistic features, (2) fine-tuning the LM on target task data to adapt it to the specific domain, and (3) fine-tuning the classifier layers for the downstream task. Novel techniques such as discriminative fine-tuning, slanted triangular learning rates, and gradual unfreezing are introduced to mitigate overfitting and catastrophic forgetting. These methods allow the model to retain previously learned knowledge while adapting to new tasks. ULMFiT achieved state-of-the-art results on six text classification tasks, including sentiment analysis and topic classification, with error reductions of 18–24% on most datasets. ULMFiT is sample-efficient, matching traditional training methods using orders of magnitude less labeled data.

Wei et al. [89] (2022) proposed **instruction tuning** as a fine-tuning approach to improve zero-shot learning capabilities in large language models (LLMs). In their study, a 137B parameter LaMDA [17] transformer model was fine-tuned on 62 NLP datasets, each reformulated into instructional templates representing tasks such as natural language inference, translation, and summarization. These templates included various natural language prompts, enabling the model to interpret task instructions effectively. To evaluate its generalization ability, entire task clusters were held during training, ensuring the model encountered unseen tasks during testing.

FLAN demonstrated superior zero-shot performance on 20 out of 25 datasets compared to GPT-3 and exceeded few-shot GPT-3 results on benchmarks like ANLI and BoolQ. The study also found that increasing the diversity of datasets and templates during fine-tuning enhanced the model's adaptability across task types.

Zhang et al. (2021) [93] examined fine-tuning BERT in scenarios with limited data, where large pre-trained models often exhibit instability. They identified several factors that contribute to suboptimal performance, such as biased gradient estimation from the widely used BERTADAM optimizer and inefficiencies in transferring knowledge from higher layers of the pre-trained model. The study demonstrated that adopting a debiased ADAM optimizer [40], selectively reinitializing the top layers in the BERT architecture, and extending the fine-tuning duration mitigated overfitting and improved model convergence. These adjustments improved stability and reduced the need for repeated trials, typically required to counteract inconsistent results. Evaluations on the GLUE dataset showed that these refinements led to more reliable fine-tuning performance across multiple benchmarks.

Another technique used to train very large language models is to target specific parameters instead of the entire model [68]. Houlsby et al. (2019) [33] tackle the challenges of fine-tuning large models such as BERT when applied to multiple tasks. They introduced adapter modules, lightweight and trainable layers that are added to a frozen pre-trained model, that drastically reduce the number of task-specific parameters required. Unlike traditional fine-tuning, which adjusts all model parameters, this method keeps the base model intact and focuses on training the adapters. These modules work effectively because they focus on modifying only the higher-level representations while preserving the robust general-purpose features learned during

pre-training. As a result, it achieves performance comparable to full fine-tuning on benchmarks such as GLUE while adding only 3.6% task-specific parameters per task. This approach is particularly useful in settings with sequential tasks, where models must adapt to new tasks without forgetting previous ones, offering a highly efficient alternative to transfer learning in NLP. However, existing techniques, such as adapter tuning, often introduce inference latency by extending the model depth or reducing the usable sequence length, posing challenges for latency-sensitive applications [35].

To address this, *LoRA* (Low-Rank Adaptation) [35] provides a more memory-efficient alternative by freezing the pre-trained model weights and injecting trainable low-rank matrices into existing layers. Unlike adapter tuning, which adds additional layers, LoRA optimizes weight updates within existing dense layers, significantly reducing the number of trainable parameters. This results in up to 10,000 times fewer parameters while maintaining strong performance on tasks like the GLUE and NLG challenges. By focusing task-specific updates on lightweight modules, LoRA minimizes GPU memory usage and training costs while avoiding additional inference latency, making it an effective solution for adapting large-scale models like GPT-3.

A more recent advancement in parameter-efficient fine-tuning is *QLoRA* (Quantized Low-Rank Adaptation) [20], which further reduces memory requirements while preserving the performance of full-precision fine-tuning. Unlike LoRA, which operates on full-precision models, QLoRA applies 4-bit quantization to pre-trained model weights, significantly lowering memory usage while allowing gradients to backpropagate through a frozen quantized model into trainable low-rank adapter matrices. This technique enables fine-tuning large-scale models, such as LLaMA-65B, on a single 48GB GPU without compromising task performance. To achieve this, QLoRA

introduces several innovations: (a) 4-bit NormalFloat (NF4), a quantization format optimized for normally distributed weights, (b) Double Quantization, which compresses quantization constants to further reduce memory overhead, and (c) Paged Optimizers, which dynamically allocate memory to prevent spikes during training. Empirical results demonstrate that QLoRA-trained models, such as the Guanaco family, achieve state-of-the-art performance on instruction-following tasks, surpassing previous open-weight models and reaching 99.3% of ChatGPT's performance on the Vicuna benchmark after only 24 hours of fine-tuning on a single GPU.

Over-specialization in fine-tuning large language models occurs when a model becomes too narrowly focused on a specific task or dataset, limiting its ability to generalize. While this can lead to higher accuracy on the fine-tuned task, it often comes at the cost of flexibility, making the model less effective in different domains. This issue arises when fine-tuning overly optimizes the model for a particular task format early in training, reinforcing patterns that may not transfer well to other applications. Studies have shown that factors such as training data order, weight initialization, and fine-tuning duration significantly affect model stability and generalization. Without careful calibration, these factors can contribute to overfitting, reducing adaptability and making the model overly reliant on the fine-tuning dataset [88].

To mitigate over-specialization, researchers have explored strategies such as early stopping, learning rate scheduling, and careful selection of fine-tuning iterations, all of which have been shown to improve model robustness across tasks [24]. More sophisticated approaches include parameter-efficient techniques like prompt tuning and LoRA, which allow models to retain general-purpose knowledge while adapting to new tasks. For example, Wang et al. [88] propose a two-stage method called Prompt

Tuning with Model Tuning (ProMoT). Since models tend to overfit to the fine-tuning task very early in training, they first apply prompt tuning, where only a small set of parameters (soft prompts) are adjusted while keeping the core model unchanged. This step helps the model capture task-specific information without losing its broader generalization ability. In the second stage, these learned prompts guide fine-tuning of the model itself, striking a balance between task specialization and cross-task adaptability. The ProMoT framework showed significant success, particularly in biomedical text summarization, where it achieved a 6.3% improvement in F1-score over conventional fine-tuning approaches. Additionally, it maintained strong performance on unrelated tasks, such as legal document classification, which demonstrates its ability to maintain generalization.

2.5 Radiology Report Summarization

The first attempt at abstractive radiology report summarization (RRS) of the *Findings* section to generate the *Impression* section was introduced by Zhang et al in 2018 [94]. Their work automated the generation of Impressions based on neural sequence-to-sequence learning with a copy mechanism similar to the one used by [76]. This early work achieved promising results for the clinical validity of NLP in radiology; where a board certified radiologist indicated that 67% of the sampled summaries generated by the [94] model were at least as good as the summaries generated by an expert.

2.5.1 Fine-Tuned LLMs in Radiology Report Summarization

In a significant advancement, Cai et al. developed *ChestXRayBERT* [10], a domain-specific pre-trained language model tailored to summarize chest radiology reports. To address the unique linguistic features of radiology texts, the authors constructed a specialized corpus of 49 million words derived from 20,617 scientific articles related to radiology. This corpus was used to pre-train a BERT-based encoder through tasks such as masked language modeling and next-sentence prediction, ensuring that the model captured domain-specific semantics and contextual relationships. The pre-trained BERT encoder was then fine-tuned on chest radiology report datasets, including MIMIC-CXR, to optimize its performance in generating concise and accurate summaries. To generate the *Impression* section from the *Findings*, the ChestXRayBERT encoder was paired with a Transformer-based decoder, forming a sequence-to-sequence architecture. When evaluated, ChestXRayBERT outperformed both general-purpose models like BART and domain-specific models like BioBERT and ClinicalBERT in terms of ROUGE and BLEU scores. Furthermore, expert evaluations confirmed its outputs as highly factual, readable, and informative. This approach has also been used in other works such as [69] where they combined a Transformer decoder with a BERT encoder pre-trained on the MIMIC-CXR dataset and then evaluated on the OPEN-I dataset to verify transferability of learned knowledge.

Building on advances in domain-specific adaptations of LLMs, Van Veen et al. proposed **RadAdapt** [85], a methodology designed to efficiently adapt LLMs for the summarization of radiology reports. Similarly to ChestXRayBERT [10], a key factor in their approach was the use of domain adaptation through pre-training on clinical text combined with lightweight fine-tuning techniques. By pre-training a T5 model on

a corpus of clinical datasets, including MIMIC-III and PubMed, the authors ensured that the model captured domain-specific semantic patterns essential for summarization tasks. For task-specific fine-tuning, RadAdapt employed LoRA which significantly reduced computational and memory requirements by only updating 0.32% of the model's parameters during fine-tuning, compared to full end-to-end training. Furthermore, LoRA preserved the original model weights, making it feasible to adapt models with hundreds of millions of parameters without extensive resources. When evaluated on the MIMIC-CXR dataset, RadAdapt with LoRA fine-tuning achieved state-of-the-art results, surpassing prior methods in BLEU (18.9), ROUGE-L (44.5), and F1-RadGraph (41.8) scores. The lightweight fine-tuning strategy further allowed the model to generalize effectively across diverse imaging modalities and anatomies.

Van et al. (2024) [87] also fine-tuned large language models (LLMs) such as GPT-4 and FLAN-T5 for clinical summarization tasks, including radiology reports, patient queries, and progress notes. They used in-context learning (ICL) and quantized low-rank adaptation (QLoRA) to adapt the models, using domain-specific datasets such as MIMIC-CXR and MeQSum. The models were evaluated through automated metrics and a clinical reader study with 10 physicians, focusing on completeness, correctness, and conciseness. GPT-4, adapted with ICL, demonstrated greater completeness over medical experts ($p < 0.001$) and generated fewer hallucinations (5% vs. 12%). While the models excelled with structured datasets, such as radiology reports, the model struggled with progress notes as they contained higher variability in vocabulary and formatting. These datasets often included unstructured data, such as free-text numerical values and inconsistent phrasing, making it harder for models to identify and summarize key information accurately.

Several other works have adapted LLMs for RRS, leveraging both fine-tuning techniques and in-context learning strategies. Tariq et al. [83] fine-tuned a T5 model with 770 million parameters on the MIMIC-III dataset using a noisy ground truth generated by LLaMA. Their approach focused on generating dual-purpose summaries: technical impressions for clinicians and simplified layperson summaries for patients, demonstrating a 63% improvement in patient comprehension compared to original impressions. Similarly, Nishio et al. [59] explored the scalability of T5 for multilingual RRS, fine-tuning the model on two large datasets: MIMIC-CXR for English reports and JMID for Japanese CT and MRI reports. Their work highlighted the impact of the diversity of the data sets, achieving ROUGE-L scores of 54.07 and 46.91 for the English and Japanese data sets, respectively, and the radiologist evaluations confirmed the clinical applicability of their summaries.

In contrast to fine-tuned models, Ma et al. [53] employed ChatGPT as a general-purpose LLM for RRS through their proposed *ImpressionGPT* framework. They used dynamic prompts to retrieve semantically similar reports from a diagnostic corpus and paired this with an iterative optimization algorithm. This approach bypassed the need for model fine-tuning, instead enhancing in-context learning to generate concise and clinically consistent impressions. On MIMIC-CXR and OpenI datasets, their framework achieved high ROUGE-L (47.93 and 65.47, respectively) and BERTScore (91.41 and 93.83, respectively) metrics, demonstrating the potential of prompt engineering and iterative refinement to adapt general-purpose LLMs for specialized tasks. A distinguishing aspect of their methodology was iterative prompt refinement through

feedback loops, which allowed the model to improve progressively based on prior outputs without extensive retraining on labeled datasets. This iterative approach, utilizing domain-specific examples, enabled *ImpressionGPT* to perform competitively with other approaches in accuracy and contextual alignment, as evidenced by evaluations on the MIMIC-CXR and OpenI datasets.

2.6 Evaluation of Machine Generated Text

Various metrics have been developed to automatically assess the quality of machine generated text. Metrics such as ROUGE [45] and BLEU [62] are widely used to quantify the textual overlap between the generated and reference summaries. A summary of common automated metrics is provided in Table 2.1.

Metric	Description
ROUGE-1, ROUGE-2, ROUGE-3	Measures the overlap of unigrams (ROUGE-1), bigrams (ROUGE-2), and trigrams (ROUGE-3) between the generated summary and the reference summary, focusing on recall.
ROUGE-L	Captures the longest common subsequence (LCS) between the generated and reference summaries, emphasizing fluency and structural similarity.
BLEU-1, BLEU-2, BLEU-3, BLEU-4	Evaluates the precision of n-gram overlaps, with higher scores indicating summaries that closely match the reference in content and style.
BERTScore	Uses embeddings from a pre-trained language model to measure semantic similarity between the generated and reference summaries.
F1-CheXbert, F1-RadGraph	Domain-specific metrics that evaluate clinical correctness by comparing the generated summaries with reference reports in terms of detected observations and anatomical entities.

Table 2.1: Common Metrics for Evaluating Summaries

The **BLEU** (Bilingual Evaluation Understudy) metric was initially developed for machine translation and later adapted for summarization. It calculates how much overlap exists between a candidate translation (or summary) and one or more reference texts, typically by counting shared n-grams. The simplest case is unigram matching—counting individual words—and while this can be informative, it sometimes inflates precision scores if a model repeats common words. To address this, BLEU introduces a concept known as **modified unigram precision**. First, it determines how often each word appears across the reference summaries, then "clips" the counts in the candidate summary to avoid overestimating matches. Formally, the modified unigram precision is given by:

$$\text{Modified Precision} = \frac{\sum_{\text{n-gram} \in \text{Candidate}} \min(\text{Count(n-gram)}, \text{MaxRefCount(n-gram)})}{\sum_{\text{n-gram} \in \text{Candidate}} \text{Count(n-gram)}} \tag{2.1}$$

This approach is then extended to higher-order n-grams, such as bigrams or trigrams, to better capture fluency in the candidate text. BLEU evaluations are usually performed at the sentence level, though they can also be aggregated across an entire dataset.

Unlike BLEU, which leans toward *precision*, the **ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) [45] library focuses more on *recall*. ROUGE compares machine-generated summaries against reference summaries written by humans and measures how effectively the model covers the essential content of the source text. This emphasis on recall makes it especially useful for summarization, where capturing all critical information is often more important than precision alone. Among

its many variants, ROUGE-N (n-gram based), ROUGE-L (longest common subsequence), ROUGE-W (weighted longest common subsequence), and ROUGE-S (skip-bigram) are the most widely used.

- **ROUGE-N** calculates the overlap of n-grams—contiguous sequences of N words—between a generated summary and the reference summaries. This metric evaluates how many of the key phrases present in the reference summaries are also included in the generated text. The formula for ROUGE-N is expressed as:

$$\text{ROUGE-N} = \frac{\sum_{\text{n-gram} \in \text{Reference}} \text{Count}_{\text{match}}(\text{n-gram})}{\sum_{\text{n-gram} \in \text{Reference}} \text{Count}(\text{n-gram})} \tag{2.2}$$

  In this formula, N represents the length of the n-gram. The numerator counts the n-grams in the reference summaries that are also present in the candidate summary, while the denominator represents the total number of n-grams in the reference summaries.

- **ROUGE-L** moves beyond individual n-grams to evaluate the **longest common subsequence** (LCS) shared by the generated and reference summaries. The LCS measures the degree of structural similarity, accounting for word order while allowing for gaps. A longer LCS between the summaries indicates a higher level of alignment in their structure and content.

- **ROUGE-W** refines ROUGE-L by assigning greater weight to **consecutive matches**, giving greater importance to longer sequences of matching phrases.

- **ROUGE-S** assesses **skip-bigram co-occurrences**, which involve pairs of words that appear in the same order in both the candidate and reference summaries, regardless of the distance between them. By analyzing non-contiguous

yet related word pairs, ROUGE-S identifies meaningful connections missed by adjacent matches.

METEOR (Metric for Evaluation of Translation with Explicit ORdering) [2] is designed to address some of the limitations of BLEU by incorporating recall alongside precision, making it a more balanced metric for evaluating text generation. Unlike BLEU, which relies solely on n-gram precision and applies a brevity penalty, METEOR directly considers how well a generated text captures the meaning of a reference by matching words based on surface forms, stems, synonyms, and even paraphrases.

METEOR assesses text similarity by aligning unigrams from the generated text to those in a reference. These alignments are determined in several stages: first matching exact words, then considering stemmed forms, and finally looking for semantically related terms (e.g., synonyms). This ensures that synonymous or slightly varied phrasing still contributes to the final score. After establishing alignments, METEOR calculates a weighted F-score that balances precision and recall, with an added emphasis on recall so the model is encouraged to preserve as much of the reference meaning as possible. Furthermore, METEOR includes a fragmentation penalty that lowers the score for text with disjointed or scrambled word order, thereby rewarding outputs that maintain coherent, natural structure. Because it accounts for both meaning and fluency, METEOR often correlates more closely with human judgment than precision-centric metrics like BLEU.

BERTScore [92] introduces an embedding-based approach to text evaluation. Unlike n-gram-based metrics, which rely on exact matches, BERTScore uses contextualized embeddings to calculate similarity. For each token in the candidate sentence,

similarity scores are computed against tokens in the reference sentence. This approach is effective at recognizing paraphrases and capturing semantic similarity.

The authors of BERTScore demonstrated its advantages over traditional n-gram methods. For example, n-gram metrics like BLEU and METEOR may favor syntactically correct but semantically inaccurate sentences. For instance, the phrases "people like visiting places abroad" and "people like foreign cars" would score higher under these metrics than the semantically correct paraphrase, "consumers prefer imported cars." BERTScore avoids such pitfalls by leveraging contextualized embeddings to capture paraphrastic relationships. Another strength of BERTScore is its ability to handle distant dependencies and reorderings. While n-gram metrics may mildly penalize the swapping of cause and effect clauses or other order changes, contextualized embeddings can effectively handle these cases.

F1-CheXbert is a metric specifically designed for evaluating the performance of systems analyzing chest X-ray radiology reports. CheXbert [80], a domain-adapted version of BERT, is used to classify chest X-ray findings into specific clinical labels. CheXbert is fine-tuned on radiology datasets to extract clinical observations, such as "pneumonia" or "effusion," and their attributes (e.g., presence, absence, or uncertainty). F1-CheXbert balances precision and recall to measure how effectively systems identify and categorize findings in chest X-ray reports. This metric is essential for evaluating the accuracy of automated tools in medical imaging analysis.

Similarly, **RadGraph** [36] is a framework designed to extract clinical entities and their relationships from free-text radiology reports, creating a structured representation of the medical information contained in the text. Its primary purpose is to evaluate how well automated systems capture clinically relevant details, such as

anatomical references and diagnostic observations, as well as the relationships between them (e.g., "Located At" or "Suggestive Of"). By focusing on the semantic and relational content of radiology reports, RadGraph provides a domain-specific evaluation tool tailored to the needs of medical applications.

The **F1-RadGraph** metric is a performance measure derived from the RadGraph framework. It assesses the precision and recall of clinical entities and their relationships identified by a system, capturing both how accurately information is extracted and how comprehensively it is represented. This metric aggregates the results into a micro F1 score for overall performance and a macro F1 score to provide insight into specific categories, such as types of errors or particular entity types. Research has shown that F1-RadGraph and BERTScore are the metrics that align most closely to radiologists. Additionally, RadGraph has been shown to align more closely with other domain-specific metrics such as CheXBert [90].

2.6.1 Evaluation of Machine Radiology Report Summarization

Evaluation approaches in recent work on LLM-based radiology summarization have primarily focused on both automated metrics and human assessments to gauge the fidelity, completeness, and clarity of generated texts. A number of studies rely on traditional n-gram overlap metrics such as **ROUGE** and **BLEU**, while others incorporate more advanced or domain-specific metrics like **BERTScore** and **F1-RadGraph**, which evaluate semantic similarity and clinical correctness respectively. For instance, Zhang et al. [94] utilized ROUGE-1, ROUGE-2, and ROUGE-L to compare generated impressions against reference summaries, complemented by a blind radiologist

evaluation of 100 sampled outputs. In addition to these standard overlaps, they investigated model robustness via cross-organization and cross-body-part tests, concluding with a detailed error analysis that categorized issues like missing critical information or ungrammatical text. Similarly, Cai et al. [10] employed ROUGE (focusing on recall) and BLEU (emphasizing precision) to measure textual overlap, coupled with expert-driven human evaluations of readability, factual correctness, and informativeness.

Several groups supplement these traditional scores with specialized or more comprehensive metrics. Van et al. [85] assessed generated summaries with ROUGE-L, BLEU, and **BERTScore**—computing semantic similarity via pre-trained contextual embeddings—as well as **F1-RadGraph**, a domain-specific measure of factual correctness based on radiology entity–relation graphs. They also conducted a radiologist reader study, scoring attributes like critical information capture and coherence on a 0–10 scale. Further highlighting the importance of factual accuracy, Tariq et al. [83] proposed tracking hallucination rates and missing information percentages, comparing a fine-tuned T5 model against a larger LLaMA model in terms of factual completeness. Their human evaluations included both laypersons—who rated understandability—and board-certified radiologists, who focused on clinical correctness, thereby capturing a wide spectrum of end-user perspectives.

Other researchers have explored correlations between automated metrics and domain expert judgments. Nishio et al. [59] reported moderate positive correlations between ROUGE-2 and radiologist ratings on a 5-point Likert scale, but they emphasized the limitation of purely n-gram-based scores, which showed considerable variance

even for summaries deemed clinically adequate. Ma et al. [53] addressed factual consistency explicitly by introducing a metric that compares disease-related observations in the generated and reference texts. They also incorporated **BERTScore** to capture semantic overlaps and performed ablation studies to isolate the contributions of iterative prompt optimization. Meanwhile, Zhu et al. [95] combined conventional metrics like BLEU, METEOR, and ROUGE-L with the CheXpert Labeler, evaluating clinical correctness across 14 predefined observations. They additionally explored the use of GPT-3.5 and GPT-4 as "quasi-human" evaluators, assigning sentence-level scores and iterating on model outputs based on LLM-driven feedback. Their final analysis compared these automated and LLM-derived scores to physician-provided ratings, finding that some large language models aligned closely with expert judgments.

Chapter 3

Methodology

This chapter describes the methodology used in this study to fine-tune a language model for radiology report summarization. The overall process, as illustrated in Figure 3.1, consists of four main phases: dataset acquisition and preprocessing, first-stage fine-tuning, second-stage fine-tuning, and evaluation.

3.1 Dataset Description

The **MIMIC-CXR dataset** [37] is a publicly available collection of chest radiographs accompanied by semi-structured free-text radiology reports. The dataset contains 377,110 images from 227,835 studies involving 65,379 patients conducted at the Beth Israel Deaconess Medical Center between 2011 and 2016. Each study includes one or more images, generally featuring frontal and lateral views. The data set is de-identified under HIPAA Safe Harbor, removing all PHI to protect patient confidentiality.

Practicing radiologists authored the radiology reports in MIMIC-CXR during routine clinical care. The reports were de-identified with rule-based methods to remove sensitive data while retaining clinical content. They are organized into plain text files

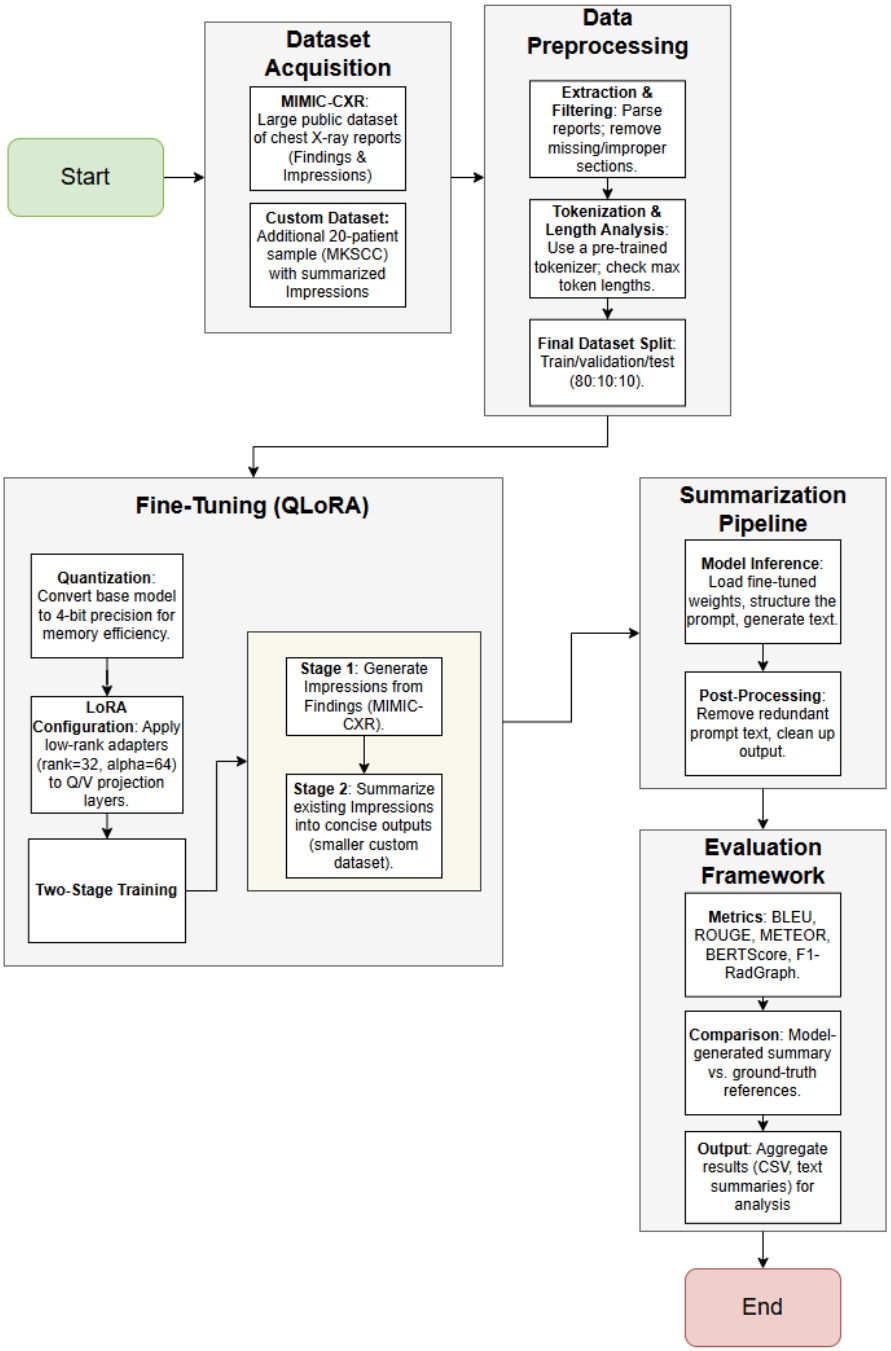

Figure 3.1: Overview of the methodology pipeline, beginning with dataset acquisition and preprocessing, quantization for LLaMa, followed by fine-tuning (LoRA), a summarization pipeline, and finally evaluation.

linked to their corresponding imaging studies via anonymized identifiers.

The dataset supports research in NLP, computer vision, and clinical data mining. Its size and detailed annotations make it ideal for developing machine learning models for radiology summarization, decision support, and diagnostics.

We also used a **custom in-house dataset** to test our model's performance against another source of radiology reports. For our custom dataset, we randomly selected 20 patients from the broader Memorial Sloan Kettering (MSK) cancer center dataset described in [23]. A radiologist summarized the list of impressions for each patient's history (totalling 174 impressions), focusing on critical elements such as trends in metastases size, diagnostic findings, temporal changes, and important observations in imaging results.

3.1.1 Data Preprocessing

Preprocessing ensures that datasets are well-structured and clean for downstream analysis and modeling. The MIMIC-CXR dataset is initially stored in a compressed format with radiology reports organized in a hierarchical directory structure. Preprocessing begins by extracting all files into a temporary directory, after which each report is parsed to identify key sections: *Background*, *Findings*, and *Impression*. Regular expressions are used to extract text from these sections, and each report is mapped to its corresponding `subject_id` and `study_id` based on the directory structure and filenames.

Once parsed, the extracted data are consolidated into a structured tabular format and saved as a CSV file with predefined headers. Reports that are missing critical sections, specifically the *Findings* or *Impression*, are excluded from the dataset to

ensure consistency with the summarization task. These sections form the input-output pairs for training and evaluation, and their absence renders a report unusable for this purpose.

The generated CSV is cleaned by removing the entries with missing or malformed information. Reports with empty fields are discarded, and additional preprocessing steps are applied, such as stripping redundant whitespace and standardizing text formatting, to ensure that the data are ready for tokenization. Of a total of 227,835 reports, 106,695 were excluded due to missing content in either or both of the required sections, resulting in a final dataset comprising 121,140 reports. The dataset is then split into training, validation, and testing subsets using an 80:10:10 ratio with stratified sampling.

Tokenization and Length Analysis

Tokenization transforms raw text into a numerical format that LLMs can interpret. This process splits text into smaller units, known as tokens, which may represent words, subwords, or characters based on the chosen tokenizer. This study uses a pre-trained tokenizer from the Hugging Face Transformers library to process the *Findings* and *Impression* sections of each radiology report. The tokenizer converts these sections into numerical sequences, which the model uses for training and inference.

Transformer-based models have a fixed input size (2048 tokens for LLaMa 3.1); meaning all inputs have to be either truncated or padded to be of the same length. We analyzed the token lengths of the dataset to determine the model constraints during training. We found that the maximum length in tokens for *Findings* and *Impression* are 394 and 289 tokens, respectively. Note that the total input includes the number of

tokens for the prompt, *Findings*, and *Impression* combined. We selected a maximum input size of 1024 tokens during model training to ensure that no inputs are truncated; instead, they are padded, allowing us to retain all information.

3.2 Fine-tuning Process

In this section, we outline the methodology used to fine-tune the base model leveraging low-rank adaptation (LoRA) for efficient parameter updates, and quantized LoRA (QLoRA) for memory-efficient training specifically in the LLaMA setup. We use two rounds of fine-tuning, first on MIMIC-CXR and then on our custom dataset. The fine-tuning process is illustrated in Figure 3.2.

We repeat this pipeline for two different architectures: a causal decoder-only model (LLaMA-3) and an encoder-decoder sequence-to-sequence model (T5). While most hyperparameters and procedures were kept consistent across both experiments, we highlight architecture-specific differences where applicable.

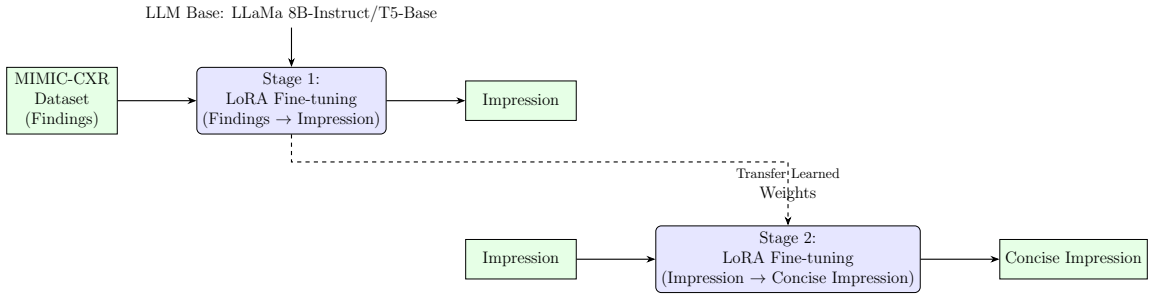

Figure 3.2: Two-stage fine-tuning process for radiology report summarization: In Stage 1, the LLM base is fine-tuned with LoRA (with quantization for LLaMa) on the MIMIC-CXR dataset to generate an *Impression* from *Findings*. In Stage 2, the model undergoes further fine-tuning to refine the *Impression* into a more concise summary.

3.2.1 Quantization and Model Preparation

For the LLaMA-based experiments, the base model is loaded using the Hugging Face `transformers` library and quantized into 4-bit precision using the BitsAndBytes configuration. We use normalized float 4 (nf4) quantization, which is the default setting and recommended by Hugging Face for improved performance [21]. Double quantization is enabled to further reduce memory overhead by approximately 0.4 bits per parameter, and computations are carried out in half-precision (`float16`) to maximize GPU utilization.

Prior to integrating QLoRA, the model is prepared for low-bit training using the `prepare_model_for_kbit_training` utility. This preparation involves casting non-trainable parameters to `float32`, enabling gradient checkpointing for the output embedding layer, and upcasting the language model head to `float32` to preserve numerical stability during training.

In contrast, the T5 model was fine-tuned using standard LoRA without quantization, as 4-bit training is currently not supported for encoder-decoder architectures in this context.

3.2.2 Targeted Layers and LoRA Configuration

LoRA is applied to specific layers of the transformer models to inject task-specific knowledge while keeping most parameters frozen. In the LLaMA experiments, we targeted the query projection (`q_proj`) and value projection (`v_proj`) submodules of the self-attention layers across all 32 transformer blocks. These components influence the generation and propagation of attention scores, and targeting both has been shown to yield strong empirical performance [35].

For T5, we extended LoRA to include the cross-attention layers in the decoder, in addition to self-attention layers, to accommodate the encoder-decoder structure and ensure effective adaptation during sequence-to-sequence training.

The LoRA configuration used the following hyperparameters:

Rank (`r`): 32

LoRA Alpha: 64

Dropout: 5%

Bias: None

Task Type: Causal Language Modeling (LLaMA)
or Sequence-to-Sequence Modeling (T5)

Rank defines the number of trainable parameters added per layer. A higher rank increases model adaptability, but also memory consumption. LoRA Alpha defines the parameter for LoRA scaling. LoRA introduces low-rank trainable matrices A and B to modify the pre-trained weight matrix W without updating all parameters. Instead of directly fine-tuning W, the LoRA update is defined as:

$$\Delta W = AB \tag{3.1}$$

To control the magnitude of these updates, a scaling factor (α) is introduced:

$$\Delta W = \frac{\alpha}{r} AB \tag{3.2}$$

where: - α is the LoRA scaling factor, - r is the LoRA rank, which determines the size of the low-rank matrices, - A and B are the learnable LoRA matrices.

The division by r ensures that the updates remain normalized and stable, preventing large deviations from the pre-trained model while still allowing effective adaptation. In this work, we set LoRA Alpha = 64 and LoRA Rank = 32, meaning the updates are scaled to balance fine-tuning efficiency and stability.

The dropout factor determines the probability of dropping out LoRA layers during training. This acts as a regularization step by allowing the model to train its parameters under varying configurations, and therefore reduces the risk of overfitting to the training data.

Finally, we set the task type to match the model architecture: causal language modeling for LLaMA (next-token prediction), and sequence-to-sequence modeling for T5 (autoregressive decoding conditioned on encoder representations).

A custom data collator is implemented to handle dynamic padding. For both model types, each batch contains sequences of uniform length by padding `input_ids` with the `pad_token_id` and masking padded labels with `-100`. The `attention_mask` is generated accordingly to exclude padded tokens from contributing to the self-attention mechanism. For encoder-decoder models like T5, the collator also ensures that decoder inputs and labels are correctly padded and aligned for sequence-to-sequence training.

3.2.3 Fine-Tuning on MIMIC-CXR: Generating Impressions from Findings

The first stage of fine-tuning was conducted on the MIMIC-CXR dataset, where the model was trained to generate radiology *Impressions* from *Findings*. The objective was to teach the model how to synthesize key abnormalities and overall interpretations

based on detailed chest X-ray findings. The training data consisted of paired *Findings-Impression* sections, formatted as follows:

```
Below is a patient's chest X-ray findings: {findings_text}

Task: Provide a concise Impression that captures the key abnormalities
and overall interpretation:
```

The fine-tuning process was conducted using the Hugging Face `Trainer` API, with a custom training loop for loss computation. The model was trained in mixed precision (`fp16`) to optimize memory usage while maintaining numerical stability. A combination of gradient accumulation, weight decay, and early stopping was used to ensure efficient optimization and stable convergence.

This procedure was applied to both the LLaMA and T5 architectures. In the case of T5, the Hugging Face trainer automatically handled sequence-to-sequence loss computation (token-level cross-entropy over decoder outputs), whereas for LLaMA, the objective was causal next-token prediction.

The training hyperparameters were configured as follows:

Batch Size:	2 (per device, for both training and evaluation)
Gradient Accumulation Steps:	4
Learning Rate:	1×10^{-5}
Warmup Steps:	100
Weight Decay:	0.01
Epochs:	10
Max Gradient Norm:	1.0
Evaluation Strategy:	Every 100 steps
Logging Steps:	Every 50 steps
Checkpointing:	Every 500 steps (max 2 checkpoints retained)
Dataloader Workers:	2
Metric for Best Model Selection:	Evaluation Loss

To prevent overfitting, early stopping was applied, where training would terminate if validation loss did not improve over 3 evaluation steps. The best-performing model checkpoint was automatically reloaded at the end of the training.

3.2.4 Fine-Tuning on Concise Impressions: Summarizing Impressions from Reports

In the second fine-tuning stage, the model was adapted to generate *concise Impressions* from full radiology *Impressions*, summarizing key findings and capturing the most relevant clinical information. Because we are using a much smaller dataset of size 174, we used **5-fold cross-validation**. This stage followed a similar training approach but with a different prompt format:

```
Task: Given the following radiology impression, summarize it
focusing on key findings and changes: {original_impression}
```

For this round of fine-tuning, the model was initialized with the weights from the first round (MIMIC-trained model) and trained on a much smaller dataset. The hyperparameters were adjusted to account for the smaller dataset size, with more frequent evaluations and checkpointing to monitor performance. The updated configuration is as follows:

Batch Size:	4 (per device, for both training and evaluation)
Gradient Accumulation Steps:	8
Learning Rate:	1×10^{-5}
Warmup Steps:	20
Weight Decay:	0.01
Epochs:	100
Max Gradient Norm:	1.0
Evaluation Strategy:	Every 25 steps
Logging Steps:	Every 25 steps
Checkpointing:	Every 50 steps (max 2 checkpoints retained)
Dataloader Workers:	2
Metric for Best Model Selection:	Evaluation Loss

Given the limited dataset size, training was extended to 100 epochs, allowing the model to make multiple passes over the data. The frequent evaluation and early stopping mechanism ensured that the model did not overfit. At the end of training, the best checkpoint was automatically selected based on evaluation loss.

3.2.5 Justification for Configuration Choices

The selection of appropriate hyperparameters is crucial for training any type of machine learning model. In the following, we provide a rationale for the chosen configuration. Although most of the choices were based on established practices in natural language processing and deep learning research, we also used manual experimentation with various settings before reaching the final configuration.

Batch Size and Gradient Accumulation A batch size of 2 was selected for the larger MIMIC-CXR dataset as any larger batches would cause an out-of-memory error. For our much smaller dataset, we used a batch size of 4. In order to achieve an effectively larger batch size without exceeding memory limits, gradient accumulation steps were set to double the batch size, which allows the model to virtually see 4 and 8 times more samples before performing a weight update, respectively. This approach has been shown to stabilize training and improve gradient estimation [60].

Learning Rate and Warmup Steps A learning rate of 1×10^{-5} was chosen based on common practices for fine-tuning large transformer-based models [22, 47]. This relatively low learning rate helps avoid catastrophic forgetting of pre-trained knowledge while allowing the model to adapt to new domain-specific tasks. The use of warmup steps (100 for the first stage, 20 for the second) provides a gradual increase in the learning rate from near zero, preventing large, destabilizing updates in the initial training phases. Since 1×10^{-5} is sufficiently small, we did not use a smaller learning rate in the second stage.

Weight Decay and Max Gradient Norm Weight decay at 0.01 helps regularize the model and prevent overfitting by penalizing large weights, following the decoupled weight decay approach in AdamW [49]. This is particularly important in smaller or domain-specific datasets to maintain generalization. A maximum gradient norm of 1.0 (gradient clipping) further stabilizes training by preventing excessively large updates, a common measure in the training of deep neural networks [63].

Epochs and Early Stopping Early stopping is widely regarded an effective method of preventing overfitting in deep learning [13]. In the first stage (MIMIC-CXR), we set the number of training epochs to 10 as the dataset is very large and we expected the model to converge earlier than the set number of epochs. In fact, the training ended after only 2.25 epochs with early stopping for the LLaMa model, and 7.7 epochs for the T5 model. In the second stage, where the dataset was significantly smaller, the number of epochs was increased to 100 to allow multiple passes over the data, helping the model capture rare patterns. However, prolonged training on smaller datasets also raises the risk of overfitting, which is why early stopping was applied again, ending training after 31.23 ± 2.79 epochs for the LLaMa model, and 50.0 ± 0.0 epochs for the T5 model. Early stopping was triggered if validation performance failed to improve three consecutive times. Validation performance was monitored frequently—every 100 steps in the first stage and every 25 steps in the second.

Choice of Cross Entropy Loss The model was optimized using Cross Entropy Loss, the standard objective for both causal and sequence-to-sequence language modeling tasks, as implemented in Hugging Face's training pipeline.

For causal decoder-only models such as LLaMA, the loss is computed over the

predicted token at each position t given all previous tokens in the sequence:

$$\mathcal{L} = -\frac{1}{N}\sum_{t=1}^{N}\log P(x_t|x_{<t}) \tag{3.3}$$

For encoder-decoder models like T5, the same cross-entropy formulation is applied over the decoder output, where the prediction of each token y_t is conditioned not only on the previous tokens $y_{<t}$ but also on the encoder representation of the input sequence.

In both settings, Hugging Face's `Trainer` efficiently computes the loss by masking padded tokens and optionally using automatic mixed precision (AMP) for performance. For LLaMA, this aligns with the pretraining objective of autoregressive next-token prediction, while for T5, it supports teacher-forced decoding in sequence-to-sequence generation.

3.3 Summarization Process

In this section, we outline the methodology used for generating model outputs for each task. We include the procedure, prompts and the generation parameters.

3.3.1 Summarization Pipeline for MIMIC-CXR

The summarization script begins by loading the fine-tuned model and its associated tokenizer. For the LLaMA-based experiments, the base model LLaMa-3.1-8B-Instruct is initialized using the Hugging Face `transformers` library. We first load the base model into memory and then apply the LoRA-adapted weights from the fine-tuned checkpoint. The tokenizer is explicitly configured to use the end-of-sequence token (`eos_token_id`) as the padding token, which is not defined by default.

For T5-based experiments, we follow the same procedure using a pre-trained encoder-decoder model t5-base), along with its tokenizer. Since T5 shares encoder and decoder embeddings and includes a predefined padding token, no manual override is required.

Once the model is loaded, inference is performed through a helper function, `generate_summary`, which takes an input text (radiology *Findings*) and generates a corresponding *Impression*. The input is first tokenized, converted to tensor format, and transferred to the model's device. Inference is conducted in a non-differentiable mode using `torch.no_grad()` to prevent unnecessary gradient calculations and reduce memory consumption.

Output generation is performed using autoregressive decoding. For LLaMA, this corresponds to left-to-right token generation without an encoder, whereas for T5, the decoder generates the output sequence conditioned on the encoder's representation of the input. A maximum token limit of 128 is set to ensure concise outputs while capturing the essential details of the *Findings*. We use a temperature of 0.7, which allows the model to explore diverse output tokens while maintaining coherence.

To maintain uniformity in how the model interprets its input, the summarization script enforces a structured prompt template. Each *Findings* entry is preprocessed by converting it to lowercase and stripping extraneous whitespace before being formatted into a predefined instruction-based prompt. This prompt explicitly directs the model to synthesize the *Impression* by extracting key abnormalities and summarizing the overall interpretation. The prompt is structured as follows:

```
Task: Provide a concise summary that captures the key abnormalities
and overall interpretation of the following findings: {findings_text}
```

After generating a raw output, the script applies a secondary post-processing step to remove redundant text. The model output often includes the original prompt within its generated output. Therefore, it must be filtered out to extract the raw *generated Impression* for performance metrics when comparing with the *target Impression*. This is achieved using the `remove_prompt_from_output` function, which identifies occurrences of the full prompt within the generated text and removes them.

The summarization script is designed to process entire datasets in batch mode. The script reads a CSV file containing chest X-ray *Findings*, iterates over each entry, and generates corresponding *Impressions*. The summarized outputs are stored in a new column labeled `Summaries` and subsequently written to a new CSV file for downstream evaluation. The final output file retains the original `Impressions` column alongside the newly generated `Summaries`.

3.3.2 Summarization Pipeline for Generating Concise Impressions

The summarization pipeline for the second stage of the fine-tuned model is nearly identical, with a few key differences in the parameters passed to the model's `generate` method. These changes were applied consistently across both the LLaMA and T5 models:

- The temperature is adjusted to 0.6, which we found to be optimal based on a range of tested values: [0.1, 0.4, 0.5, 0.6, 1].
- The model was configured with top-k = 50 and top-p = 0.9, using nucleus sampling to restrict token selection to a subset of highly probable words. This modification improves response fluency while maintaining diversity.

- Unlike the previous task, which relied primarily on stochastic sampling, this script incorporates beam search (`num_beams=10`) to enhance determinism and refine output coherence.
- A length penalty of 0.3 was introduced to encourage shorter outputs.
- The repetition penalty was lowered to 1.1 to penalize redundant patterns.

Instead of instructing the model to synthesize *Findings* into an *Impression*, the input now explicitly directs it to extract key information and observed changes from an existing *Impression*. The new prompt format is as follows:

```
Task: Given the following radiology impression, summarize it
focusing on key information and observed changes: {original_impression}
```

3.4 Model Selection and Architecture

The models selected for this study are **LLaMa 3.1-8B-Instruct** and **T5-base**. LLaMa 3.1-8B-Instruct represents the smallest version of the LLaMa 3.1 family of large language models. This choice was based on its strong performance in benchmark evaluations, as reported in the LLaMa 3.1 paper [48]. LLaMa 3.1-8B-Instruct consistently outperformed other models of comparable size (8B parameters) in key tasks, making it well-suited for fine-tuning radiology report summarization on our Nvidia A100/40GB GPU. Although the LLaMa 3.1-70B and LLaMa 3.1-405B versions significantly outperform our chosen model, using such large models is not feasible on our computing resources.

3.4.1 LLaMa 3.1-8B-Instruct

Traditionally, encoder–decoder transformers (like BART or T5) have been favored for summarization because the encoder can thoroughly process the input text. However, modern decoder-only models can effectively internalize the input context within their self-attention mechanism. When summarizing with a decoder-only model, we simply prepend the radiology report (or its findings section) as part of the prompt, possibly with a separator and an instruction. The model attends to the entire prompt (which includes the input text) and then generates the summary. This approach has proven not only viable but in some cases highly competitive with encoder–decoder models. For example, GPT-4's summary performance on clinical text has been so strong that it often rivals specialized models; in radiology, GPT-4's summaries were as clinically accurate as expert-written ones [87]. Furthermore, decoder-only models have been shown to encode substantial domain-specific knowledge and exhibit strong reasoning abilities in clinical applications [79]. This study explores whether a small decoder-only model such as LLaMa 3.1-8B-Instruct, when fine-tuned on radiology report data, can achieve competitive or superior performance compared to traditional encoder-decoder models.

LLaMa 3.1-8B-Instruct performed strongly in both general and domain-specific benchmarks, including MMLU [31] (Massive Multitask Language Understanding). MMLU is a well-established benchmark that evaluates model performance across 57 subjects, spanning STEM fields (e.g., mathematics, physics, computer science) and humanities (e.g., history, law, philosophy). It measures a model's ability to tackle complex reasoning tasks that demand both factual knowledge and problem-solving skills. LLaMa 3.1-8B-Instruct achieved 73.0% accuracy on the MMLU benchmark,

outperforming GPT-3.5 Turbo (69.8%) and Gemma 2-9B (72.3%).

In GSM8K [16] reasoning tasks, which test grade-school-level math problems, LLaMa 3.1-8B achieved a score of 84.5%, indicating it solved the problems effectively without need for chain-of-thought (CoT) prompting, a technique often required for smaller models. Other models of comparable size, such as Gemma 2-9B and Mistral 7B, achieved scores of 76.6% and 53.2%, respectively. This suggests that LLaMa 3.1-8B-Instruct has a strong internal representation of logical and quantitative reasoning, which makes it highly effective for structured summarization tasks such as the processing of radiology reports.

The decision to utilize LLaMa 3.1-8B-Instruct is further supported by its fine-tuning methodology tailored for instruction-based tasks. This adaptation makes the model ideal for summarizing specialized content, such as radiology reports. Unlike generic LLMs, the instruction-based fine-tuning of LLaMa 3.1 ensures improved contextual comprehension and adherence to domain-specific guidelines. Practical factors like computational efficiency and accessibility were also considered. Its smaller size (8B parameters) balances performance and resource efficiency, making it more practical for research compared to larger models.

LLaMa 3.1 Architecture

LLaMa 3.1 [48] improves upon LLaMa 2 with better efficiency, scalability, and support for long-form text. Its architecture is built on a dense Transformer model, simplifying training and inference by removing the *expert* component division. Table 3.1 summarizes the key components of the LLaMa 3.1-8B architecture.

Feature	Description
Grouped Query Attention (GQA)	Uses eight key-value attention heads to reduce memory usage and speed up decoding, while still maintaining the model's ability to capture long-range dependencies.
Segmented Attention Masking	Ensures that different documents in a packed training batch remain independent, preventing interference and improving model performance on longer sequences.
Expanded Vocabulary	The tokenizer now supports around 128,000 tokens—100,000 from a core set and 28,000 for non-English languages—resulting in better tokenization and improved multilingual capabilities.
Extended Rotary Positional Embedding (RoPE)	Increases RoPE frequency to 500,000, allowing the model to process sequences longer than 32,000 tokens without losing attention stability.
Eight-Layer Transformer Architecture	Comprises eight transformer layers, each with multi-head attention and feedforward networks, enabling deeper feature extraction for better understanding of syntax and meaning.

Table 3.1: Key Architectural Features of LLaMa 3.1-8B [48]

Pre-training Strategy of LLaMa 3.1

The LLaMa-3.1 language model is initially trained on a large percentage of the textual data available on the web. To ensure the quality of the data, filters are used to exclude data from websites containing unsafe content (such as adult or hateful content) and high volumes of personally identifiable information [48].

First, each raw text sample undergoes a series of preprocessing steps, including extraction from source formats, removal of intrusive markup, and normalization to plain text. Next, redundancies are elimnated through de-duplication at multiple scales (e.g., document, line, and URL) to minimize repeated content. In addition, domain-targeted heuristics are employed to exclude problematic text segments, such as those

containing unsafe or personal information. Model-based quality filtering methods are used to help further refine the corpus, enabling the removal of low-quality documents and the preservation of high-value tokens in both general and specialized domains [48].

After data set curation, classifiers are used to analyze the distribution of different data domains, such as natural language, mathematical reasoning, and multilingual text, to determine their relative contributions to overall downstream performance. Meta uses proxy models, smaller-scale versions of the target model, and trains them on different compositions of the dataset to predict the behavior of the target model. This iterative approach allows Meta to predict the optimal dataset composition for the larger models without expending excessive computing resources. In some cases, annealing is applied in later training stages, where the model is fine-tuned on select high-quality data (e.g., mathematical reasoning) to improve performance on specific benchmarks. The goal is to improve specific capabilities while maintaining a balanced representation across domains [48].

Post-Training Strategy of LLaMa 3.1

The final version of the LLaMa 3.1 model goes through several rounds of post-training that incorporate human feedback. Each round of post-training includes supervised fine-tuning (SFT), in which the pre-trained model is trained on curated interaction data containing prompts paired with target response [48]; similar to the fine-tuning strategy used in this thesis.

In addition to supervised fine-tuning, reward modeling (RM) and direct preference optimization (DPO) help adjust the model's behavior to better match human

expectations. The reward model is trained on preference data, where human annotators or model-based evaluations rank multiple candidate responses from best to worst. Unlike traditional methods that rely only on binary classification (e.g., chosen vs. rejected responses), the dataset also includes "edited" responses, which improve selected answers through minor refinements [48].

At this stage, the DPO loss function is structured to avoid excessive optimization on formatting tokens, which could cause abrupt cutoffs, repetitive text, or unstable responses. To address this, special formatting tokens, such as header markers, message delimiters, and instruction separators, are masked during the loss computation, keeping the optimization process centered on meaning rather than token structure. A regularization term based on negative log-likelihood (NLL) loss stabilizes formatting and preserves the model's ability to generate well-structured responses [48].

Although Reinforcement Learning with Human Feedback (RLHF) has been a common approach for preference alignment, recent studies suggest that direct preference optimization (DPO) improves computational efficiency and training stability, especially in large models [48]. DPO eliminates the need for complex reinforcement learning frameworks such as PPO, instead directly optimizing the policy model based on human-labeled preference data. Studies indicate that DPO outperforms RLHF on instruction-following benchmarks like IFEval and does so with lower computational costs. Similarly to the pre-training stage, late-stage annealing is applied to fine-tune the model on specialized datasets for mathematical reasoning and complex problem solving. [48].

3.4.2 T5-Base

The inclusion of T5-base, an encoder-decoder model, serves as a valuable point of comparison against the decoder-only architecture of LLaMa 3.1-8B-Instruct. T5-base is an encoder-decoder transformer model developed explicitly for text-to-text tasks, making it suitable for summarization tasks. Its architecture allows for distinct processing stages, where the encoder fully contextualizes the input before the decoder generates the output. This separation often translates to strong summarization capabilities, particularly when dealing with structured, information-dense text like radiology reports.

Moreover, T5-base's encoder-decoder structure explicitly separates context encoding from output generation, enabling it to focus intensely on understanding and condensing input text during summarization. This characteristic potentially enhances its ability to distill key clinical insights from radiology reports, providing a clear contrast to the context-embedding strategies employed by decoder-only models. By directly comparing these two fundamentally distinct architectures—encoder-decoder versus decoder-only—our study aims to offer comprehensive insights into their relative strengths, weaknesses, and applicability in clinical radiology summarization tasks.

T5-Base Architecture

The T5 (Text-to-Text Transfer Transformer) model introduced by Google [68] employs an encoder-decoder Transformer architecture that unifies a wide variety of language tasks under a single text-to-text framework. Rather than training separate models for tasks like translation, classification, or summarization, T5 reformulates each task into a text-in, text-out paradigm. In other words, all inputs (e.g., source

sentences, classification queries) and outputs (e.g., translated sentences, class labels) are represented as text strings, thereby streamlining training and inference across diverse natural language processing applications.

Table 3.2 highlights the main components of T5-Base, which has roughly 220 million parameters and strikes a balance between performance and computational efficiency. The model uses a multi-headed self-attention mechanism in both the encoder and the decoder, along with feedforward networks that capture contextual dependencies within the sequence. Additionally, layer normalization and skip connections help stabilize training while enabling deeper representations of text.

Feature	**Description**
Encoder-Decoder Transformer	Uses a stacked encoder to produce contextual embeddings of the input, and a stacked decoder to generate the output tokens in an autoregressive manner.
Shared Embeddings	Both encoder and decoder share the same token embedding parameters, promoting efficient parameter usage and consistent token representations.
Span-Corruption Objective	Pre-training masks contiguous spans of tokens (rather than individual tokens), which improves the model's understanding of phrasal and semantic coherence.
Relative Positional Encoding	Employs relative positions instead of absolute indices, enabling the model to better generalize across sequences of varying lengths.
12-Layer Transformer Blocks	T5-Base adopts 12 layers each in the encoder and decoder, with 12 attention heads and feedforward networks to capture complex linguistic patterns.

Table 3.2: Key Architectural Features of T5-Base [68]

Pre-training Strategy of T5

Like many large language models, T5 is initially pre-trained on a massive dataset—in this case, the Colossal Clean Crawled Corpus (C4) [68], which is derived from extensive web text but filtered and cleaned to reduce undesirable content and increase text quality. A main component of T5's pre-training is the "span-corruption" or "span-masking" objective, where contiguous sequences of tokens in the input are masked and the model is trained to predict the missing spans. This approach encourages the model to learn deeper semantic and syntactic information by forcing it to generate chunks of text, rather than isolated tokens.

Before feeding data into the model, the text undergoes preprocessing, which includes normalization and removal of noise (e.g., HTML markup), similar to the de-duplication strategy employed in LLaMa 3.1 [48]. Additionally, filtering stages discard documents containing highly repetitive text or sensitive information. This refined corpus ensures that the model encounters fewer noisy samples, thus improving downstream performance. By covering a wide range of tasks (e.g., translation, classification, question answering) in a single text-to-text objective, T5 effectively transfers its language understanding to a variety of domains.

Post-Training Strategy of T5

After pre-training, T5 undergoes task-specific fine-tuning on labeled datasets. Since T5 natively handles all tasks in a text-in, text-out format, these downstream tasks merely require formulating the input-output pairs to match T5's expected format. For instance, in sentiment classification, the input might be prefixed with a task identifier (e.g., *"sst2 sentence:"*), followed by the raw text. The model then outputs

a sentiment label (*"positive"* or *"negative"*) in textual form. This unified schema simplifies both the training pipeline and the application of T5 to new tasks.

While supervised fine-tuning on labeled data is the most common post-training strategy, further domain adaptation is possible if large unlabeled corpora are available. Techniques such as continued pre-training or multi-task learning can be employed to refine the model's linguistic capabilities for specialized use-cases, such as biomedical text or legal documents [68]. As with the LLaMa 3.1 approach [48], annealing or curriculum learning may be introduced in later stages of training to emphasize high-value tokens in niche domains, helping to maintain a balanced representation across different knowledge areas.

3.5 Evaluation Framework

The evaluation pipeline begins by extracting the model-generated summaries and their corresponding ground-truth references from a test dataset, unseen by the model during training. The evaluation framework is identical for both models (MIMIC fine-tuned and two-stage fine-tuned).

During preprocessing, both references and predictions are converted to lowercase to ensure uniformity in evaluation. Additionally, extraneous whitespace is removed, and tokenization is applied where required for specific metrics. The processed texts are then passed to various evaluation modules, each designed to measure a different aspect of the summarization quality. The metrics used are **BLEU, ROUGE, METEOR, BERTScore, and F1-RadGraph**. Please refer to the **Related Work: Evaluation and Quality Assessment of LLM-Based Summarization in Radiology** section for a detailed description of these metrics.

Chapter 4

Results

The **Results** section is structured to present both quantitative and qualitative outcomes of the two-stage fine-tuning approach. Training was carried out sequentially for two model architectures—`LLaMa 3.1-8B-Instruct` and `T5-base`. Each model was first fine-tuned on the MIMIC-CXR dataset to generate radiology *Impressions* from *Findings*, followed by a second fine-tuning phase aimed at producing *concise impressions* from original radiology impressions using our custom in-house dataset.

The performance of each fine-tuning phase was monitored by tracking the training loss, validation loss, and validation perplexity (PPL) throughout the training steps.

For both models, training and validation losses were computed using the cross-entropy loss function. In the case of LLaMA, the loss was computed autoregressively, where the loss at each token position t was determined by the negative log-likelihood of the token given all previous tokens:

$$\mathcal{L} = -\frac{1}{N}\sum_{t=1}^{N} \log P(x_t | x_{<t}) \tag{4.1}$$

Where N represents the number of tokens in the sequence. For T5, the same

cross-entropy loss was applied over the decoder output, where each predicted token was conditioned on the encoder's input and previously generated decoder tokens.

Perplexity, a commonly used measure of uncertainty in language modeling, was calculated from the validation loss. It reflects how well the model predicts a sample and is derived from the average cross-entropy loss $\mathcal{L}$ using the standard transformation:

$$\text{Perplexity} = e^{\mathcal{L}} \tag{4.2}$$

Lower perplexity values indicate greater model confidence and improved predictive performance. The exponential is used because perplexity is interpreted as the effective number of choices the model has when making predictions, transforming the average loss into a more interpretable quantity on the same scale as the vocabulary size.

Following an assessment of training and validation performance, we evaluate each model's summarization quality using established metrics: BLEU, ROUGE, METEOR, BERTScore, and F1-RadGraph. These results are then compared with benchmarks reported in the literature to assess relative model performance.

4.1 Training Results

The fine-tuning of the models across both tasks was monitored using training loss (e.g 10.0), validation loss (e.g 3.0), and validation perplexity (e.g 1.2) throughout the training steps. Training and validation loss measure how well the model fits the training and validation data. Validation perplexity measures the uncertainty of the value of a sample.

4.1.1 Training Results For The MIMIC-CXR Task

LLaMa Training Results

Figure 4.1 shows the training loss, validation loss, and perplexity over steps throughout the training process. *Training loss* showed a steep decrease in the initial training phase, dropping from approximately 20 to below 10 within the first hundred steps. This rapid decrease suggests that the model quickly adapted to the task-specific data distribution. The loss reduction rate slowed as training progressed, stabilizing around 7.5–8 after 4000 steps. The training loss fluctuations in the later steps are normal and indicate that the model was continuously updating its parameters.

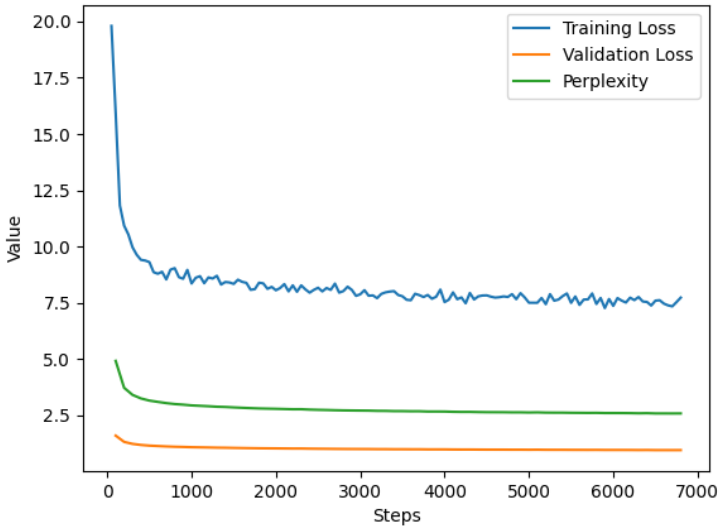

Figure 4.1: Training loss, validation loss, and perplexity over steps for the LLaMa model fine-tuned on the MIMIC-CXR task.

The *validation loss* followed a similar downward trend, starting at approximately 1.6 and steadily decreasing to just below 1.0. The smooth decline suggests effective

generalization to unseen validation samples, with no sudden spikes indicative of overfitting. The narrowing gap between training and validation loss toward the end of training further demonstrate that the model was not memorizing training data but learning meaningful summarization representations.

The *validation perplexity* strongly correlated with the validation loss. Perplexity was initially near 5.0, indicating a high uncertainty level in the model's predictions. As training progressed, perplexity consistently decreased and eventually converged to a value slightly above 2.5. This indicates that the model became increasingly confident in its generated output.

Interestingly, the validation loss is lower than the training loss. To validate the observed training dynamics, we analyzed the composition of the training and validation splits to ensure that no structural biases or data leakage contributed to the low validation loss. A cross-check of subject–study pairs confirmed that there were zero duplicates between the training and validation sets, ruling out any direct overlap that could inflate evaluation performance. Additionally, we examined the average impression length and found that both splits were nearly identical: the training set had a mean of 15.22 tokens and a median of 9, while the validation set had a mean of 15.24 tokens and a median of 9. This parity suggests that the validation samples were not shorter or inherently easier to summarize. Moreover, the vocabulary richness, measured using type-token ratio, was consistent across the splits (0.074 for training and 0.073 for validation), further indicating comparable linguistic complexity. Given these findings, the slightly lower validation loss is most plausibly explained by regularization mechanisms (such as dropout and LoRA dropout) that are active only during training, as well as potential noise in the training loss curve due to its computation

on partial batches. These effects, together with the absence of overfitting signals, suggest that the model maintained strong generalization throughout training.

T5 Training Results

Figure 4.2 shows the training loss, validation loss, and perplexity over steps for the T5 model during fine-tuning on the MIMIC-CXR dataset. *Training loss* exhibited a rapid decline early in training, falling from approximately 3.4 to below 2.0 within the first 500 steps. This early improvement suggests that the model quickly adapted to the task-specific distribution of chest X-ray reports. After the initial sharp decrease, the training loss continued to decline gradually, stabilizing around 1.4 toward the final training steps. Minor fluctuations observed in later stages are expected and reflect ongoing adjustments to model parameters.

The *validation loss* mirrored the training loss trend, decreasing from an initial value of around 2.5 to below 1.3 by the end of training. The consistent downward trajectory and absence of significant spikes indicate effective generalization, with no signs of overfitting. The narrowing gap between training and validation losses suggests the model was learning meaningful semantic mappings between the *Findings* and *Impression* sections without overfitting to the training distribution.

Validation perplexity followed a similarly smooth curve, beginning at a high value near 12.5 and decreasing steadily throughout training. By the final steps, perplexity had converged to approximately 3.7, reflecting the model's increasing confidence in predicting coherent and contextually accurate summaries. This reduction in perplexity confirms that the model was able to effectively leverage the encoder-decoder architecture to capture phrasal and semantic relationships in the MIMIC-CXR data.

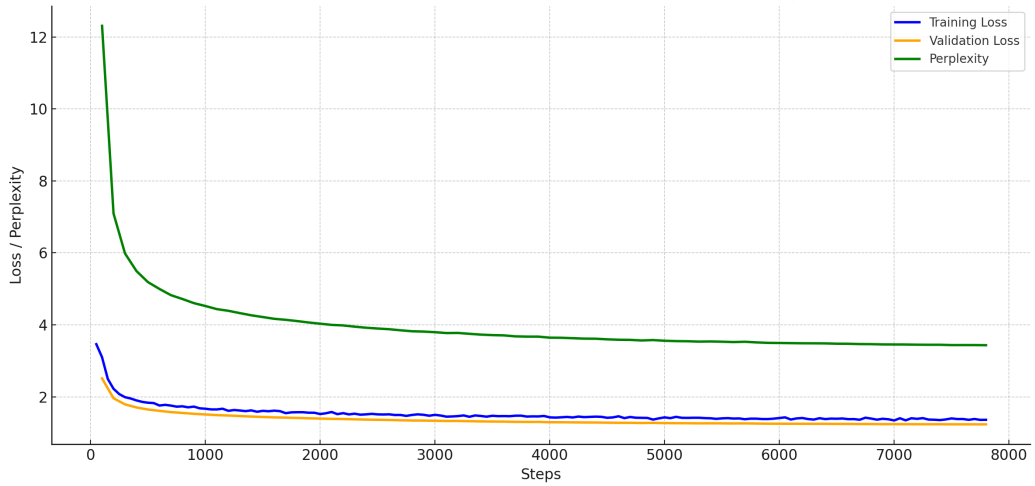

Figure 4.2: Training loss, validation loss, and perplexity over steps for the T5 model fine-tuned on the MIMIC-CXR task.

4.1.2 Training Results For The Impression Summarization Task

LLaMa Training Results

Given the limited size of the dataset, a **5-fold cross-validation** approach was used to train and evaluate the model across multiple data splits. The band in the graphs represents the standard deviation. This approach was not feasible on the MIMIC-CXR task due to required training time associated with a very large dataset.

Figure 4.3 shows the mean training loss, validation loss, and perplexity across the training steps, while the shaded regions indicate standard deviation. The *mean training loss* decreased rapidly in the initial steps, dropping from approximately 14.5 to below 4.0 within the first 125 steps. This sharp decline indicates that the model quickly adapted to the task, learning to refine its generated impressions. As training progressed, the loss continued to decrease at a slower rate, stabilizing around 2.0. The relatively narrow standard deviation band suggests consistency across folds, indicating

that the model's training behavior remained stable despite different validation splits.

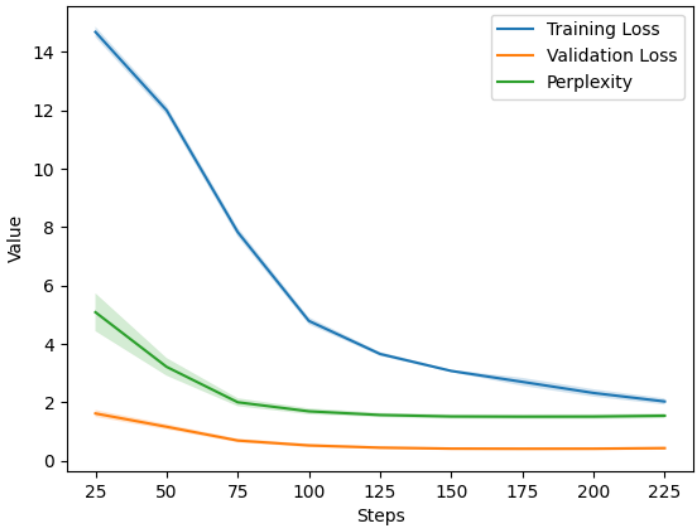

Figure 4.3: Training loss, validation loss, and perplexity over steps for the LLaMa model on the concise impression generation task (5-fold cross-validation). The solid lines represent the mean values, and the shaded regions represent the standard deviation..

The *mean validation loss* started at approximately 1.6 and exhibited a consistent downward trend, reaching a minimum of around 0.4 before slightly increasing towards the end. This is taken into account by loading the optimal checkpoint of the model prior to slight degradation in performance. The standard deviation bands indicate moderate variability in validation loss between different data splits, which is expected given the small dataset size.

Initially, the model training rounds exhibited high *mean validation perplexity* (≈ 5.0), which indicates substantial uncertainty in its generated summaries. However, perplexity decreased rapidly during early training, stabilizing below 2.0 in later

stages. This pattern suggests that the model successfully improved its ability to generate fluent and structured concise impressions. Similarly, the standard deviation was relatively large initially but steadily decreased throughout training, indicating that all five folds converged to a similar level of performance.

T5 Training Results

Figure 4.4 presents the mean training loss, validation loss, and perplexity across training steps, with shaded regions denoting standard deviation. The *mean training loss* exhibited a consistent decline, beginning around 1.1 and dropping below 0.9 by the final steps. This trend indicates steady improvements in the model's ability to summarize full impressions into concise versions. The narrow standard deviation bands throughout training suggest high consistency in optimization behavior across the five cross-validation folds.

The *mean validation loss* followed a similar downward trajectory, decreasing from approximately 1.3 to just above 1.0 by the end of training. The smooth decline and stable deviation bands indicate reliable generalization performance across folds, with little variability introduced by the different validation subsets.

The *mean validation perplexity* started at around 3.5, reflecting moderate uncertainty in early generations. As training progressed, perplexity declined steadily, converging to a value near 2.7. This improvement indicates the model's increasing confidence in producing fluent, clinically meaningful summaries. The narrowing shaded region toward the final steps suggests that the training procedure yielded consistent convergence behavior across all folds.

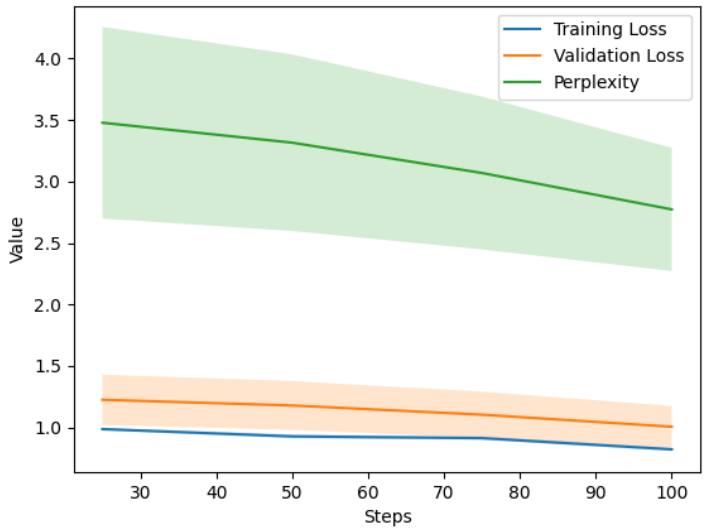

Figure 4.4: Training loss, validation loss, and perplexity over steps for the T5 model on the concise impression generation task (5-fold cross-validation). The solid lines represent the mean values, and the shaded regions represent the standard deviation.

4.2 Evaluation Results

Evaluating medical text summarization requires metrics that go beyond surface-level word overlap. Traditional metrics such as BLEU and ROUGE are widely used in summarization tasks but may not fully capture semantic similarity or clinical accuracy in radiology reports. BLEU-4, for instance, measures the exact match of n-grams and often penalizes paraphrased but clinically equivalent summaries. ROUGE-L similarly focuses on longest common subsequences, which may misrepresent the quality of abstractive outputs that omit redundant findings or rephrase conclusions. While these metrics provide a baseline for syntactic alignment, they can be misleading when used in isolation, particularly for domain-specific tasks that prioritize clinical meaning over lexical overlap.

To address these limitations, we incorporated BERTScore and F1-RadGraph, two metrics more aligned with the objectives of radiology report summarization. BERTScore evaluates the contextual similarity of embeddings between generated and reference texts which captures semantic relationships even when wording differs. F1-RadGraph measures overlap in clinical entities and relations based on a structured graph representation, making it particularly suitable for assessing the medical fidelity of the output.

4.2.1 Performance on the MIMIC-CXR Dataset

Performnce of LLaMa

The first stage of fine-tuning involved adapting LLaMa 3.1-8B-Instruct on the MIMIC-CXR dataset to generate radiology *Impressions* from *Findings*. The model demonstrated strong performance on this task, achieving a BLEU-4 score of 28.26 (174.18% increase over base model), ROUGE-L of 46.29 (115.57% increase over base model), BERTScore of 87.52 (16.08% increase over base model), and F1-RadGraph of 39.01 (111.15% increase over base model) as shown in Table 4.1.

However, the introduction of a second round of fine-tuning on a custom dataset led to a significant decline in performance on the original MIMIC task, with BLEU-4 and ROUGE-L scores dropping to 6.41 (125.05% decrease) and 20.31 (78.02% decrease), respectively. The BERTScore also declined from 87.52 to 80.78 (8.01% decrease), indicating that while the model retained some level of semantic similarity to reference texts, it exhibited increased divergence in content. Most notably, the RadGraph score saw a substantial decrease from 39.01 to 13.24 (98.64% decrease), indicating a significant decline in the medical information similarity between the *Findings* and

Model	B-1	B-2	B-3	B-4	R-1	R-2	R-L	M	BS	RG
Base LLaMa-3.1-8B	8.55	4.70	3.05	1.95	11.97	4.70	9.60	21.03	74.49	11.14
MIMIC-CXR Fine-Tuned LLaMa-3.1-8B	**44.44**	36.36	31.79	28.26	49.18	35.88	46.29	**48.26**	**87.52**	39.01
Two-Stage Fine-Tuned LLaMa-3.1-8B	22.58	12.65	8.90	6.41	23.72	10.46	20.31	26.91	80.78	13.24
T5-Base	14.85	7.79	5.54	4.02	15.55	6.59	13.39	21.52	77.28	9.30
MIMIC-CXR Fine-Tuned T5-Base	43.62	**37.40**	**33.91**	**31.25**	**49.27**	**38.52**	**47.57**	46.66	87.46	**40.73**
Two-Stage Fine-Tuned T5-Base	43.39	37.17	33.69	31.04	49.22	38.54	47.60	46.42	87.39	40.55

Table 4.1: Evaluation Metrics for MIMIC-CXR Task. Scores are reported in percentages. **B** = BLEU, **R** = ROUGE, **M** = METEOR, **BS** = BERTScore, **RG** = RadGraph.

Impression.

This degradation in F1-RadGraph performance can be attributed to the summarization style of our *Impression summarization* dataset. Many examples in the dataset focused on omitting less important details from the impression while preserving only the most clinically relevant information, leading to a more concise output as shown in the first row of Table 4.2. This behavior is different from the *Findings* to *Impression* task, which involves writing entirely new text in the generated output.

dataset	Input	Target Output
Impression summarization dataset	since november 5, 2016, small bowel obstruction with transition point at the ventral hernia, similar appearance to the prior small bowel obstruction due to the incarcerated hernia. findings were discussed by dr. evan lustbader with dr. judy dong by phone on december 28, 2016 at 7:15 pm. final report	since november 5, 2016: small bowel obstruction with transition point at the ventral hernia, similar to prior due to the incarcerated hernia.
MIMIC-CXR Dataset	A right IJ central venous catheter projects over the SVC. There is no pneumothorax. Despite low lung volumes, there is no consolidation or pleural effusion. Lungs are grossly clear. Heart and mediastinum are within normal limits. Regional bones and soft tissues are unremarkable.	No intrathoracic source of infection identified. Clear lungs.

Table 4.2: Example Summarizations Across Tasks. These examples from each dataset highlight the differences in summarization style required for each task. The **impression summarization task** demonstrates the similarities between the *input text* and the *target output text*. Fine-tuning the model on this dataset degrades the F1-RadGraph score for the MIMIC-CXR dataset. The text in green highlights common text between the input and the target output.

To verify whether the two-stage fine-tuned model was aligning its summaries more closely with the original *Findings* rather than the *Impressions*, we ran the F1-RadGraph algorithm on the *generated Impressions* and compared them to the *Findings* rather than the *reference Impressions*. The results were as follows:

- **Average Entity-Level F1:** 33.16
- **Average Relation-Level F1:** 29.67
- **Average Overall F1:** 25.68

These results indicate a higher structural similarity between the generated summaries and the original *Findings*, reinforcing the observation that the two-stage fine-tuned model retained a more direct mapping from *Findings* to *Impressions*. This suggests that while the model excelled in generating concise, high-relevance summaries, its tendency to omit auxiliary details impacted its performance on structured information retention metrics such as F1-RadGraph.

Performance of T5

The first stage of fine-tuning the `t5-base` model resulted in significant increases in performance. The MIMIC-CXR fine-tuned T5-Base demonstrated strong summarization capabilities, outperforming the baseline in all evaluation metrics. As shown in Table 4.1, the model achieved a BLEU-4 score of 31.25 (677.36% increase over base), ROUGE-L of 47.57 (255.18% increase), and F1-RadGraph of 40.73 (337.31% increase), reflecting both lexical fluency and improved clinical information preservation. BERTScore also improved significantly, reaching 87.46 (13.18% increase), confirming the semantic similarity between generated and reference summaries.

Following the second round of fine-tuning on the Impression summarization dataset, the T5 model retained its strong performance on the original MIMIC-CXR task. The two-stage fine-tuned model achieved nearly identical results to the single-stage model, with only minor differences across all metrics. This suggests that fine-tuning on a

related summarization task did not interfere with the model's ability to generalize back to the original objective.

Notably, the two-stage T5 model outperformed the two-stage LLaMa model across all metrics, including BLEU-4, ROUGE-L, and RadGraph. This may be due to the encoder-decoder architecture of T5, which is naturally well suited to summarization tasks and may have benefited from the consistent structure and clinical language shared between the two datasets. These results reinforce the idea that when tasks are well-aligned in objective and format, sequential fine-tuning can serve to reinforce, rather than disrupt, the model's learned behavior.

4.2.2 Performance on the Impression Summarization Task

Performnce of LLaMa

The model fine-tuned on MIMIC-CXR not only demonstrated superior performance on the MIMIC task but also outperformed the base model on the Impression summarization task. Despite never being explicitly trained on the Impression dataset, the MIMIC-CXR fine-tuned model achieved a BLEU-4 score of 25.13 (208.72% increase over base model), ROUGE-L of 43.33 (107.52% increase over base model), BERTScore of 85.05 (7.95% increase over base model), and F1-RadGraph of 35.80 (11.28% increase over base model). These results indicate that the model transferred knowledge gained from the *Findings-to-Impression* generation task in MIMIC-CXR, allowing it to generalize well to a different dataset with similar summarization objectives. The improvement in F1-RadGraph further suggests that the fine-tuned model preserved more medically relevant entities and relations compared to the base model, reinforcing the effectiveness of domain-specific pretraining before adaptation to new

tasks.

The standard deviation across five-fold cross-validation for the *Impression summarization* task is presented in Table 4.4 due to size limitations.

Despite the observed decline in generalization to the MIMIC task, the second round of fine-tuning yielded substantial improvements in the Impression summarization task. The model exhibited a BLEU-4 score of 45.33, a ROUGE-L of 64.55, and a BERTScore of 90.75—outperforming the single-stage fine-tuned model across all metrics with a 57.34%, 39.34%, and a 6.48% increase in performance, respectively. Furthermore, F1-RadGraph scores increased from 35.80 to 43.64 (19.74% increase). This suggests that fine-tuning on Impression summarization dataset helped the model align its summaries more closely with the radiologist written summaries.

Notably, the two-stage fine-tuned LLaMa model outperformed the corresponding T5 model on the Impression summarization task across all metrics. This performance gap may be partially attributed to LLaMa's larger parameter count, which allows it to better model subtle linguistic and clinical patterns—even when fine-tuned on relatively small datasets. The increased model capacity appears to enable deeper representation learning and more flexible adaptation, helping it to align closely with the radiologist-written summaries. While T5-Base is well-suited for task-specific generation, LLaMa's scale and open-ended generative architecture likely offer an advantage in capturing nuanced domain language, even in low-resource settings.

Model	B-1	B-2	B-3	B-4	R-1	R-2	R-L	M	BS	RG
Base LLaMa-3.1-8B	14.75	11.79	10.00	8.14	23.13	16.22	20.88	47.05	78.78	32.17
MIMIC-CXR Fine-Tuned LLaMa-3.1-8B	37.76	32.44	28.79	25.13	47.51	34.28	43.33	55.24	85.05	35.80
Two-Stage Fine-Tuned LLaMa-3.1-8B	**57.91**	**52.97**	**49.31**	**45.33**	**66.05**	**55.72**	**64.55**	**73.90**	**90.75**	**43.64**
T5-Base	40.65	35.00	30.54	26.65	47.88	37.64	45.54	61.44	85.11	31.55
MIMIC-CXR Fine-Tuned T5-Base	38.24	31.53	27.17	23.16	45.56	32.52	42.40	53.19	83.96	30.63
Two-Stage Fine-Tuned T5-Base	50.84	44.73	40.28	36.01	57.35	45.42	54.69	64.14	87.71	34.06

Table 4.3: Evaluation Metrics for Impression Task. Scores are reported in percentages. **B** = BLEU, **R** = ROUGE, **M** = METEOR, **BS** = BERTScore, **RG** = RadGraph

Model	B-1	B-2	B-3	B-4	R-1	R-2	R-L	M	BS	RG
Base LLaMa-3.1-8B	1.57	1.28	1.12	1.02	2.39	1.77	1.85	1.35	1.10	5.81
MIMIC-CXR Fine-Tuned LLaMa-3.1-8B	3.48	3.24	3.21	3.06	3.28	4.20	2.69	5.17	0.97	5.25
Two-Stage Fine-Tuned LLaMa-3.1-8B	5.35	5.73	5.85	5.99	4.08	4.86	4.03	5.22	1.22	8.58
T5-Base	7.89	7.55	7.37	6.81	7.78	8.15	7.93	7.31	4.13	6.40
MIMIC-CXR Fine-Tuned T5-Base	5.19	4.81	4.48	3.91	4.98	4.96	4.41	5.74	1.38	6.27
Two-Stage Fine-Tuned T5-Base	7.65	7.04	6.71	6.38	6.37	6.34	6.14	6.24	6.30	2.96

Table 4.4: Cross-Validation Standard Deviation for Impression Task. **B** = BLEU, **R** = ROUGE, **M** = METEOR, **BS** = BERTScore, **RG** = RadGraph

Analysis of Model Behavior

To further investigate the model's summarization performance, we analyzed the distribution of concise impressions in the dataset. A clear imbalance emerged: the phrase *"no evidence of disease"* accounted for **20.23%** of all reference summaries. While this reflects real-world trends—many radiology reports document normal findings—it also means that the model was repeatedly exposed to this phrase during training, increasing the likelihood of it being overused in generated impressions.

Removing these cases to balance the dataset was not a viable option. With a relatively small dataset and a LLM, reducing the number of training examples would risk degrading generalization. Moreover, because radiology reports naturally contain class imbalances, artificially adjusting the data distribution could introduce unintended biases. Given these considerations, all training examples were retained.

To better understand how this affected model behavior, we manually examined generated outputs and identified three recurring patterns: cases where the model successfully summarized the findings, cases where it produced incorrect or misleading outputs, and cases where it copied the original impression verbatim rather than generating a concise summary. Table 4.5 presents representative examples.

The first example in Table 4.5 shows a case where the model produced a well-formed summary. The original impression contained multiple findings, references to prior imaging, and follow-up recommendations. The reference concise impression distilled the key information while omitting unnecessary details. The model's output closely aligned with this reference, demonstrating an ability to generate clinically relevant summaries.

Original Impression	Reference Concise Impression	Generated Impression
1. Since February 1, 2016, new nonspecific right lower lobe punctate nodule, attention on follow-up recommended. 2. Lesions in segment 8 and segment 7 seen on MRI from July 6, 2016 is not identified on CT. See report of PET CT performed the same day for integrated summary.	Since February 1, 2016, new right lower lobe punctate nodule, indeterminate.	Since February 1, 2016, new nonspecific right lower lobe punctate nodule.
1. Since October 19, 2013, new soft tissue focus along the inferior tip of the liver possibly peritoneal implant, possibly postoperative. 2. New thickening of the peritoneal reflection in the deep left hemipelvis possibly carcinomatosis. PET scan is suggested for further evaluation.	Since October 19, 2013, interval surgery with new possible peritoneal implant along the liver. New thickening of the peritoneal reflection in the deep left hemipelvis possibly carcinomatosis.	Since October 19, 2013, no evidence of disease.
1. Since April 08, 2016, metastatic disease in the right axilla and surrounding region has increased. 2. No substantial change in appearance of bone metastases. Final report.	Since April 08, 2016, increased right axillary metastatic disease and unchanged bone metastases.	Since April 08, 2016, metastatic disease in the right axilla and surrounding region has increased. No substantial change in appearance of bone metastases.

Table 4.5: LLaMa Qualitative Output Examples. These examples demonstrate correct behavior, wrong output, and verbatim copying of the input, in order from top to bottom.

By contrast, the second example illustrates a failure case where the model defaulted to *"no evidence of disease"* despite the presence of significant findings. Given the phrase's high frequency in the training data, the model appears to have overlearned it as a default response. While appropriate for cases without abnormalities, this kind of error could lead to critical misinterpretations in real-world applications.

The final example highlights an issue where the model did not summarize at all but instead reproduced the original impression almost verbatim. While factually

correct, this output fails to achieve the intended goal of abstraction. The tendency to copy rather than condense suggests that the model may rely too heavily on lexical similarity when uncertain about what to remove.

These findings indicate that while the model is effective in many cases, it is prone to two specific failure modes: overuse of highly frequent phrases and occasional failure to abstract information. Addressing these issues could involve refining the loss function to penalize excessive repetition or incorporating additional training objectives to encourage more selective summarization.

Performance of T5

On the Impression summarization task, the `t5-base` model demonstrated strong performance across all metrics. Notably, the base T5 model outperformed all other baseline models, achieving a BLEU-4 score of 26.65, ROUGE-L of 45.54, and BERTScore of 85.11. This aligns with the T5 architecture's strengths in sequence-to-sequence modeling, particularly in tasks requiring structured and concise generation.

Fine-tuning on the MIMIC-CXR dataset led to a slight decline in performance on the Impression task, with BLEU-4 decreasing to 23.16, ROUGE-L to 42.40, and F1-RadGraph to 30.63. While this version of the model was not directly optimized for Impression summarization, its output still retained a high degree of fluency and medical relevance, suggesting that knowledge acquired during Findings-to-Impression training remained partially transferable.

Following fine-tuning on the Impression dataset, the two-stage T5 model achieved substantial performance gains across all metrics. It reached a BLEU-4 score of 36.01

(35.16% increase over the base model), ROUGE-L of 54.69 (20.09% increase), METEOR of 64.14 (4.39% increase), and BERTScore of 87.71 (3.05% increase). Additionally, F1-RadGraph improved from 31.55 to 34.06 (7.96% increase), reflecting a higher fidelity to the clinical content in reference summaries.

These findings highlight T5's ability to adapt effectively when fine-tuned on a targeted dataset, particularly when the second task reinforces rather than diverges from the first. Unlike LLaMa, which benefited more from cross-task generalization, T5 excelled when directly exposed to task-specific supervision. This further supports the idea that encoder-decoder models can perform robustly in specialized summarization tasks, provided they are fine-tuned with representative examples that reflect the structure and tone of the desired output.

Zero-Shot Performance of T5-Base on the Impression Summarization Task

One noteworthy result is the strong zero-shot performance of the T5-Base model on the Impression summarization task. Without any prior exposure to radiology-specific data or impression summaries, the base T5 model outperformed all other baselines and even surpassed the MIMIC-fine-tuned LLaMa model. Specifically, it achieved a BLEU-4 score of 26.65, ROUGE-L of 45.54, and BERTScore of 85.11 (see Table 4.3).

This strong performance can be attributed to the stylistic alignment between the task and the T5 model's pretraining setup. Since T5 is instruction-tuned for a wide range of summarization tasks, it excels at condensing longer text into more focused summaries using extractive phrasing, even without domain-specific adaptation. In

many examples from our dataset, the required summary transformation closely resembled the type of summarization T5 was originally trained for—preserving phrasing with minor structural reordering. Table 4.6 presents an illustrative example that highlights this alignment.

Original Impression	**Target Impression**	**T5-Base Generated**
since december 8, 2017, radical cystoprostatectomy with ureteral ileal conduit and resection of bladder associated mass without ct evidence of local recurrence. right iliac lymphadenopathy is new, suspicious for focus of metastatic disease. differential however includes postsurgical reactive lymph node or, less likely, complex lymphocele.	since december 8, 2017, radical cystoprostatectomy with ureteral ileal conduit and resection of bladder associated mass. new right iliac lymphadenopathy, suspicious for focus of metastatic disease.	ct evidence of local recurrence is absent after radical cystoprostatectomy. right iliac lymphadenopathy is new, suspicious for focus of metastatic disease.

Table 4.6: T5 Zero-Shot Alignment Example. This example from the Impression summarization dataset shows strong alignment between the task and the base T5 model's generation style. Highlighted segments show content that closely matches the reference target impression.

These results suggest that instruction-tuned models like T5-Base may already possess strong capabilities in clinical summarization when the target output style is extractive or semi-extractive in nature. However, they also demonstrate the potential for overgeneration or content repetition, which can be mitigated through task-specific fine-tuning.

4.3 Comparison with Related Work

To better understand how our fine-tuned **LLaMa 3.1-8B-Instruct** model compares to other models trained for **Findings-to-Impression summarization**, we analyzed its performance against previous works using the **MIMIC-CXR** dataset. Table 4.7 presents the results based on the **ROUGE** metric, which is the most commonly reported measure across these studies.

Model	ROUGE-1	ROUGE-2	ROUGE-L
T5-base Model [59]	**57.75**	**49.96**	**54.07**
ImpressionGPT [53]	54.45	34.50	47.49
LLaMa 3.1-8B-Instruct (Our model)	49.18	35.88	46.29
Zhang et al. Model [94]	48.56	35.25	47.06
LexRank [53]	18.1	7.47	16.87

Table 4.7: Comparison to Published Models. Comparison of ROUGE scores between our fine-tuned LLaMa 3.1-8B-Instruct model and other published models for radiology report summarization (*Findings* to *Impression*) on the MIMIC-CXR dataset.

Our model achieved a **ROUGE-1 score of 49.18, ROUGE-2 of 35.88, and ROUGE-L of 46.29**, making it competitive with existing approaches but slightly behind state-of-the-art models. The **T5-base model** by Nishio et al. [59] reached the highest performance across all ROUGE scores, with **ROUGE-L 54.07**, outperforming our model by approximately **8 points**. The T5-base model demonstrated superior performance in radiology report summarization due to its encoder-decoder architecture, which is inherently optimized for text-to-text tasks. Unlike decoder-only models (such as LLaMa 3.1) that rely on autoregressive generation, T5 processes input and output sequences jointly, allowing for more effective content structuring and information retention. Additionally, its ROUGE-optimized training ensured that fine-tuning was explicitly aligned with evaluation metrics, enhancing lexical overlap and

coherence between generated and reference Impressions.

Similarly, ImpressionGPT outperformed LLaMa 3.1-8B-Instruct despite lacking task-specific fine-tuning. This advantage can be attributed to its significantly larger scale—ChatGPT, the underlying model of ImpressionGPT, consists of over 175 billion parameters, vastly exceeding the 8 billion parameters of LLaMa 3.1. More importantly, ImpressionGPT leveraged optimization during inference through an iterative refinement approach, dynamically adjusting its outputs based on contextual feedback. This process allowed for progressive quality enhancement without the need for additional training, leading to improved alignment between generated summaries and reference texts.

Compared to **Zhang et al.'s model** [94], which achieved **ROUGE-1 = 48.56** and **ROUGE-L = 47.06**, our model performed similarly, with a slightly higher **ROUGE-2 score (35.88 vs. 35.25)**. The difference in architectures likely explains these variations—our model is a **transformer-based LLM**, while Zhang et al. used an early version of the transformer architecture.

Finally, when compared to **LexRank**, a graph-based extractive summarization method, our model demonstrated a clear advantage. LexRank achieved **ROUGE-1 = 18.1, ROUGE-2 = 7.47, and ROUGE-L = 16.87**, significantly lower than all transformer-based approaches. This demonstrates the imitations of extractive summarization for radiology reports, where the transition from **Findings to Impression** is inherently **abstractive**, requiring rewording and interpretation rather than simple extraction of key sentences.

Chapter 5

Discussion and Conclusions

In this thesis, we set out to explore the feasibility of using relatively small language models—LLaMa 3.1 8B-Instruct and T5-Base—to summarize radiology reports. Specifically, we investigated whether a two-stage fine-tuning approach could effectively align the models' outputs with radiologist-produced Impressions, even when using a custom dataset of limited size. Our results confirm that LLMs, when properly adapted, can be competitive in generating concise, clinically relevant summaries—even in highly specialized medical domains.

A key takeaway is that smaller *decoder-only* LLMs can hold their own against more established encoder-decoder architectures like T5, and even significantly larger GPT-based models under certain conditions. While T5-based [59] and GPT-based [53] solutions still outperformed our models in absolute terms, we demonstrated that this gap can be meaningfully narrowed through domain adaptation and efficient fine-tuning strategies. According to our findings, encoder-decoder models such as T5 remain better suited for structured summarization tasks due to their separation of encoding and decoding pathways. However, causal decoder-only models like LLaMa

offer flexibility and open-endedness that make them well-suited to multi-purpose clinical NLP tasks beyond summarization. Through parameter-efficient fine-tuning techniques (e.g., QLoRA for LLaMa and LoRA for T5) and a two-stage training protocol, we successfully aligned both models to radiologist-generated summaries, with LLaMa demonstrating strong generalization and T5 excelling in task-specific adaptation.

Moreover, the two-stage fine-tuning strategy proved reliable when training data were limited. By initially adapting each model on a large public dataset (MIMIC-CXR) for the Findings-to-Impression task, we enabled both models to internalize radiology-specific language structures and summarization patterns at scale. We then transferred this foundational knowledge to a much smaller custom dataset of 174 Impressions. Our results showed that both LLaMa and T5 could further refine their outputs to match radiologist writing style without the need for thousands of additional samples. While the two-stage T5 model achieved the best performance on the original MIMIC-CXR task, the two-stage LLaMa model outperformed T5 on the Impression summarization task. This suggests that T5 was especially effective at preserving task-specific capabilities from the first fine-tuning stage, whereas LLaMa's larger parameter count may have offered an advantage on the smaller downstream dataset due to its greater capacity for fine-grained adaptation.

These findings carry broader implications for integrating AI-driven tools into clinical workflows. When models can be fine-tuned to capture both general medical language and localized documentation patterns with limited data, the technology becomes more accessible to smaller hospitals or specialized departments. Clinicians could curate small, targeted datasets that reflect institutional norms and reporting conventions, and then fine-tune general-purpose models to align with their specific

needs. Importantly, this approach is not limited to radiology: the same methodology can be extended to other forms of structured or semi-structured medical text such as pathology, operative, or discharge reports, where section-based summarization or impression synthesis is required.

Integrating LLMs into clinical workflows offers considerable potential to enhance medical documentation, support clinical decision-making, and improve interdisciplinary communication. Foundational models—without task-specific fine-tuning—can be leveraged to assist in note-taking during patient visits, where structured prompting can help extract main ideas, organize content logically, or identify potential inconsistencies in the discussion. In future applications, retrained models may be exposed to completely unseen reporting domains (e.g., dermatopathology, cardiology notes), yet still benefit from initial fine-tuning on adjacent medical tasks such as radiology, due to the shared linguistic and structural features common across clinical narratives.

More complex tasks, such as summarization, question-answering, or clinical decision support, require explicit fine-tuning to meet medical standards. Summarizing a radiology report, for example, requires consistent inclusion of clinically salient information. While large foundational models understand summarization in principle, they may not reliably identify which medical details to retain. Fine-tuning LLMs on domain-specific datasets allows for more predictable behavior, which is critical in high-stakes environments like healthcare. When combined with lightweight tuning techniques, these models can be rapidly adapted to new document types with minimal annotation effort, enabling generalizability across departments and institutions.

Beyond individual use cases, LLMs offer the potential to standardize medical documentation at scale. Variability in clinical text affects the clarity of patient records

and impedes interoperability across institutions. By training and deploying fine-tuned models, healthcare organizations could converge toward a more unified documentation style—characterized by consistent terminology, structure, and level of detail. Such standardization would benefit not only clinicians but also downstream AI applications by improving the quality, reliability, and interpretability of medical text. Over time, the widespread adoption of LLM-based documentation tools may enable more structured, efficient, and transparent communication in clinical medicine.

5.1 Advice for Clinical Use

Although the present work demonstrates that relatively small LLMs can approach state-of-the-art performance, it is critical to emphasize that these models should serve as *assistive* rather than autonomous tools in clinical practice. In routine radiology reporting, any omission or hallucination in the *Impression* section may have direct patient-care consequences. Our error analysis revealed that these models could produce incorrect outputs. Consequently, we recommend an interactive workflow in which the radiologist drafts the Impression and then invokes the model to (i) flag potentially missing findings and (ii) suggest corroborating phrases with explicit provenance links.

5.2 Limitations

One of the primary limitations of this study was the computational cost associated with training large-scale language models. Although our fine-tuned model demonstrated strong performance in summarizing radiology reports, the available hardware limited the extent of model improvement. We conducted our experiments using NVIDIA A100 40GB VRAM GPUs. These GPUs are powerful but insufficient for training flagship models such as LLaMA 3.1-405B from scratch. Training such ultra-large-scale models requires, at a minimum, access to 8 Nvidia H100 GPUs with 80GB VRAM each [26]. Consequently, our approach relied on parameter-efficient fine-tuning methods such as QLoRA and quantization to adapt pre-trained models without exceeding memory limitations. Although these techniques allowed us to fine-tune an 8B-parameter model effectively, they restrict the extent to which a model can be optimized for domain-specific tasks. Larger base models trained without quantization would offer superior reasoning and summarization capabilities, but our hardware constraints prevented us from directly experimenting with such models.

Another major challenge in this study was the limited size of our custom-made dataset. Unlike large-scale datasets such as MIMIC-CXR, which contain thousands of radiology reports, our dataset is comparatively small, posing a risk of overfitting and limiting the fine-tuned model's generalizability. To mitigate this issue, we implemented a five-fold cross-validation strategy, training the model five times with distinct training, validation, and test sets. We applied standard cross-fold validation techniques, splitting each iteration into 80% for training, 10% for validation, and 10% for testing, ensuring that every sample was part of the test set exactly once. We increase our test data size from 17 to 87 by averaging the performance results across

five distinct models.

Cross-validation helped improve the statistical relevance of our results, but it does not fully address the limitations of a small dataset. Language models typically benefit from exposure to diverse training examples, and the restricted sample size can prevent the model from fully capturing rare clinical patterns. Future work could benefit from large publicly available datasets on the *Impression summarization task* or explore data augmentation, synthetic text generation, or active learning strategies to expand the dataset and improve model generalization.

Given both computational and data constraints, a key trade-off in our study was selecting a small-sized model (8B parameters) rather than a much larger architecture. This choice allowed us to train and fine-tune our model within available resources efficiently. Smaller models are also better suited for small datasets, such as those used in our second round of fine-tuning. However, it limits the upper bound of performance improvements achieved with larger-scale models. Our model's relatively small size affects its performance, particularly with long inputs. In contrast, larger models, like the LLaMa 3.1 405B, demonstrate significantly greater capability in reasoning and effectively managing complex or lengthy prompts. This enhanced performance makes larger models more suitable for tasks such as summarizing extensive text or incorporating few-shot examples within the prompt.

5.3 Future Work

Using *Retrieval-Augmented Generation* (RAG) models, which integrate LLMs with external information retrieval systems, can further improve the utility of LLMs in clinical settings. RAG models can access electronic health records, medical databases,

and sources of clinical guidelines. Typically, their databases use embeddings, mathematical representations of text that encode meaning into dense vector spaces. When a query is made, such as a physician searching for similar cases or a patient seeking relevant literature, the model converts the input into a search vector and compares it against a precomputed database of embeddings. This allows for efficient and quick similarity matching for retrieving relevant information. In practice, this could enable clinicians to instantly retrieve records of patients with similar symptoms, previous cases with comparable imaging findings, or clinical guidelines relevant to the condition at hand.

Incorporating multimodal data, including both textual and imaging inputs, will further enhance clinical reasoning and decision-making. By jointly processing radiology reports, imaging studies, laboratory values, and patient histories, multimodal models can move beyond simple summarization tasks and begin to synthesize cross-modal evidence for diagnostic support. For example, models that analyze both CT images and the accompanying clinical notes could identify inconsistencies, track disease progression, or flag potential misinterpretations. This convergence of structured and unstructured data into a unified reasoning framework opens up new possibilities for context-aware, image-grounded medical inference.

Lastly, semantic routing mechanisms allow a single system to alternate its behavior based on the task provided, such as summarizing, evidence-based recommendation, or diagnostic reasoning. This allows the underlying architecture to run specialized scripts or adjust the underlying system role solely based on the semantic similarity between a given user prompt and a list of designed routes, further specializing the role of a single model to various tasks.

Bibliography

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[2] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In Jade Goldstein, Alon Lavie, Chin-Yew Lin, and Clare Voss, editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[3] Yehoshua Bar-Hillel. Some linguistic problems connected with machine translation. *Philosophy of Science*, 20(3):217–225, July 1953.

[4] Yehoshua Bar-Hillel. The future of machine translation. *Times Literary Supplement: Freeing the Mind*, (5), March-June 1962.

[5] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[6] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[7] Adrian P. Brady. Error and discrepancy in radiology: inevitable or avoidable? *Insights into Imaging*, 8(2):171–182, 2017.

[8] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[9] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[10] Xiaonan Cai, Sha Liu, Jing Han, Liang Yang, Zhi Liu, and Ting Liu. Chestxraybert: A pretrained language model for chest radiology report summarization. *IEEE Transactions on Multimedia*, 25:845–855, 2023.

[11] Jianpeng Cheng and Mirella Lapata. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association*

for Computational Linguistics (ACL), pages 484–494, Berlin, Germany, August 7–12 2016. Association for Computational Linguistics.

[12] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[13] HongSeok Choi, Dongha Choi, and Hyunju Lee. Early stopping based on unlabeled samples in text classification. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 708–718, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[14] Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.

[15] Sumit Chopra, Michael Auli, and Alexander M. Rush. Abstractive sentence summarization with attentive recurrent neural networks. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, San Diego, California, June 2016. Association for Computational Linguistics.

[16] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro

Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

[17] Aaron Daniel Cohen, Adam Roberts, Alejandra Molina, Alena Butryna, Alicia Jin, Apoorv Kulshreshtha, Ben Hutchinson, Ben Zevenbergen, Blaise Hilary Aguera-Arcas, Chung ching Chang, Claire Cui, Cosmo Du, Daniel De Freitas Adiwardana, Dehao Chen, Dmitry (Dima) Lepikhin, Ed H. Chi, Erin Hoffman-John, Heng-Tze Cheng, Hongrae Lee, Igor Krivokon, James Qin, Jamie Hall, Joe Fenton, Johnny Soraker, Kathy Meier-Hellstern, Kristen Olson, Lora Mois Aroyo, Maarten Paul Bosma, Marc Joseph Pickett, Marcelo Amorim Menegali, Marian Croak, Mark Díaz, Matthew Lamm, Maxim Krikun, Meredith Ringel Morris, Noam Shazeer, Quoc V. Le, Rachel Bernstein, Ravi Rajakumar, Ray Kurzweil, Romal Thoppilan, Steven Zheng, Taylor Bos, Toju Duke, Tulsee Doshi, Vincent Y. Zhao, Vinodkumar Prabhakaran, Will Rusch, YaGuang Li, Yanping Huang, Yanqi Zhou, Yuanzhong Xu, and Zhifeng Chen. Lamda: Language models for dialog applications. In *arXiv*. 2022.

[18] Jean-Benoit Delbrouck, Pierre Chambon, Zhihong Chen, Maya Varma, Andrew Johnston, Louis Blankemeier, Dave Van Veen, Tan Bui, Steven Truong, and Curtis Langlotz. RadGraph-XL: A large-scale expert-annotated dataset for entity and relation extraction from radiology reports. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12902–12915, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[19] 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, 2009.

[20] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[21] Tim Dettmers and Hugging Face Team. 4-bit quantization of transformers: The power of bitsandbytes, 2023. Accessed: 2025-02-05.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[23] Richard K. G. Do et al. Patterns of metastatic disease in patients with cancer derived from natural language processing of structured ct radiology reports over a 10-year period. *Radiology*, 301(1):115–122, Aug. 2021.

[24] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305, 2020.

[25] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. Why Does Unsupervised Pre-training Help Deep Learning? *Journal of Machine Learning Research*, 11(19):625–660, 2010.

[26] Hugging Face. Llama 3: Hugging face inference endpoints, 2024. Accessed: 2025-02-12.

[27] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Volume 32*, pages 41–62, July 1998.

[28] Mahak Gambhir and Vishal Gupta. Deep learning-based extractive text summarization with word-level attention mechanism. *Multimedia Tools and Applications*, 81(15):20829–20852, June 2022.

[29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[30] Jiatao Gu, Zhaopeng Lu, Hang Li, and Victor OK Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, 2016.

[31] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

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

[33] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019.

[34] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[35] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[36] Saahil Jain, Ashwin Agrawal, Adriel Saporta, Steven Truong, Du Nguyen Duong, Tan Bui, Pierre Chambon, Yuhao Zhang, Matthew P. Lungren, Andrew Y. Ng, Curtis Langlotz, and Pranav Rajpurkar. Radgraph: Extracting clinical entities and relations from radiology reports. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[37] Alistair E. W. Johnson, Tom J. Pollard, Seth J. Berkowitz, Nathaniel R. Greenbaum, Matthew P. Lungren, Chih-ying Deng, Roger G. Mark, and Steven Horng. Mimic-cxr, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific Data*, 6(1):317, Dec 2019.

[38] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall, 3rd edition, 2023.

[39] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*, 2022. Corrected publication 2022.

[40] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[41] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc.

[42] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

[43] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[44] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for*

Computational Linguistics, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[45] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[46] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China, November 3–7 2019. Association for Computational Linguistics.

[47] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Ro{bert}a: A robustly optimized {bert} pretraining approach, 2020.

[48] AI@Meta Llama Team. The llama 3 herd of models. July 2024. A detailed contributor list can be found in the appendix of this paper.

[49] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[50] Shuting Lou, Jiarui Deng, and Shanxiang Lyu. Chaotic signal denoising based on simplified convolutional denoising auto-encoder. *Chaos, Solitons & Fractals*, 161:112333, 2022.

[51] Andrew Lukaszewicz, Joseph Uricchio, and Grygori Gerasymchuk. The art of the radiology report: Practical and stylistic guidelines for perfecting the conveyance

of imaging findings. *Canadian Association of Radiologists Journal*, 67(4):318–321, 2016.

[52] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[53] Chong Ma, Zihao Wu, Jiaqi Wang, Shaochen Xu, Yaonai Wei, Zhengliang Liu, Fang Zeng, Xi Jiang, Lei Guo, Xiaoyan Cai, Shu Zhang, Tuo Zhang, Dajiang Zhu, Dinggang Shen, Tianming Liu, and Xiang Li. An iterative optimizing framework for radiology report summarization with chatgpt. *IEEE Transactions on Artificial Intelligence*, 5(8):4163–4175, 2024.

[54] Neil Macdonald. Language translation by machine — a report of the first successful trial. *Computers and Automation*, 3(2), February 1954.

[55] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[56] John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI Magazine*, 27(4):12, Dec. 2006.

[57] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In Dekang Lin and Dekai Wu, editors, *Proceedings of the 2004 Conference on Empirical*

Methods in Natural Language Processing, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[58] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gülçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In Stefan Riezler and Yoav Goldberg, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics.

[59] Mizuho Nishio, Takaaki Matsunaga, Hidetoshi Matsuo, Munenobu Nogami, Yasuhisa Kurata, Koji Fujimoto, Osamu Sugiyama, Toshiaki Akashi, Shigeki Aoki, and Takamichi Murakami. Fully automatic summarization of radiology reports using natural language processing with large language models. *Informatics in Medicine Unlocked*, 46:101465, 2024.

[60] Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor, editors, *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[61] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[62] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[63] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[64] Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, 2018.

[65] Lawrence Rabiner and B.-H Juang. *Speech Recognition: Statistical Methods*, pages 1–18. 12 2006.

[66] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[67] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018.

[68] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[69] Rujeena Reang, Vasudev Dehalwar, and R. K. Pateriya. Automatically generate impression from findings in radiology report using bert. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–6, 2024.

[70] Pengjie Ren, Zhumin Chen, Zhaochun Ren, Furu Wei, Jun Ma, and Maarten de Rijke. Leveraging contextual sentence relations for extractive summarization using a neural attention model. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, page 95–104, New York, NY, USA, 2017. Association for Computing Machinery.

[71] Andrew B Rosenkrantz, Laura Chaves Cerdas, Danny R Hughes, Michael P Recht, Sharyl J Nass, and Hedvig Hricak. National trends in oncologic diagnostic imaging. *J Am Coll Radiol*, 17(9):1116–1122, July 2020.

[72] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[73] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods*

in Natural Language Processing, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[74] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition, 2020.

[75] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, November 1997.

[76] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, 2017.

[77] Rebecca L. Siegel, Kimberly D. Miller, and Ahmedin Jemal. Cancer statistics, 2023. *CA: A Cancer Journal for Clinicians*, 73(1):17–48, 2023.

[78] Shubham Singh, Sarvnaz Karimi, Khin Ho-Shon, and Len Hamey. Show, tell and summarise: learning to generate and summarise radiology findings from medical images. *Neural Computing and Applications*, 33(13):7441–7465, Jul. 2021.

[79] Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Agüera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and

Vivek Natarajan. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, August 2023.

[80] Akshay Smit, Saahil Jain, Pranav Rajpurkar, Anuj Pareek, Andrew Ng, and Matthew Lungren. Combining automatic labelers and expert annotations for accurate radiology report labeling using BERT. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1500–1519, Online, November 2020. Association for Computational Linguistics.

[81] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS)*, pages 3104–3112. Curran Associates Inc., 2014.

[82] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[83] Amara Tariq, Sam Fathizadeh, Gokul Ramaswamy, Shubham Trivedi, Aisha Urooj, Nelly Tan, Matthew T. Stib, Bhavik N. Patel, and Imon Banerjee. Patient centric summarization of radiology findings using large language models. *medRxiv*, February 2024.

[84] Alan Turing. Computing machinery and intelligence. *Mind*, 59(236):433–60, 1950.

[85] Dave Van Veen, Cara Van Uden, Maayane Attias, Anuj Pareek, Christian Bluethgen, Malgorzata Polacin, Wah Chiu, Jean-Benoit Delbrouck, Juan Zambrano Chaves, Curtis Langlotz, Akshay Chaudhari, and John Pauly. RadAdapt: Radiology report summarization via lightweight domain adaptation of large language models. In Dina Demner-fushman, Sophia Ananiadou, and Kevin Cohen, editors, *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 449–460, Toronto, Canada, July 2023. Association for Computational Linguistics.

[86] 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.

[87] Dave Van Veen, Cara Van Uden, Louis Blankemeier, Jean-Benoit Delbrouck, Asad Aali, Christian Bluethgen, Anuj Pareek, Malgorzata Polacin, Eduardo Reis, Anna Seehofnerová, Nidhi Pontes, Poonam Rohatgi, William Hosamani, Neera Collins, Curtis P. Ahuja, Jason Langlotz, Sergios Hom, John Gatidis, and Akshay S. Pauly. Adapted large language models can outperform medical experts in clinical text summarization. *Nature Medicine*, 30(4):1134–1142, April 2024.

[88] Author Wang et al. Two-stage fine-tuning: A method to balance specialization and generalization in large language models. *Journal of Machine Learning Research*, 24(1):1–35, 2024.

[89] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models

are zero-shot learners. In *International Conference on Learning Representations*, 2022.

[90] Feiyang Yu, Mark Endo, Rayan Krishnan, Ian Pan, Andy Tsai, Eduardo Pontes Reis, Eduardo Kaiser Ururahy Nunes Fonseca, Henrique Min Ho Lee, Zahra Shakeri Hossein Abad, Andrew Y. Ng, Curtis P. Langlotz, Vasantha Kumar Venugopal, and Pranav Rajpurkar. Evaluating progress in automatic chest X-ray radiology report generation. *Patterns*, 4(9), September 2023. Publisher: Elsevier.

[91] Haoyu Zhang, Jingjing Cai, Jianjun Xu, and Ji Wang. Pretraining-based natural language generation for text summarization. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 789–797, Hong Kong, China, November 3–4 2019. Association for Computational Linguistics.

[92] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.

[93] Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. Revisiting few-sample {bert} fine-tuning. In *International Conference on Learning Representations*, 2021.

[94] Yuhao Zhang, Daisy Yi Ding, Tianpei Qian, Christopher D. Manning, and Curtis P. Langlotz. Learning to summarize radiology findings. In Alberto Lavelli,

Anne-Lyse Minard, and Fabio Rinaldi, editors, *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 204–213, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[95] Qingqing Zhu, Xiuying Chen, Qiao Jin, Benjamin Hou, Tejas Sudharshan Mathai, Pritam Mukherjee, Xin Gao, Ronald M. Summers, and Zhiyong Lu. Leveraging Professional Radiologists' Expertise to Enhance LLMs' Evaluation for AI-generated Radiology Reports . In *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, pages 402–411, Los Alamitos, CA, USA, June 2024. IEEE Computer Society.

[96] Émile Delavenay and Katharine M. Delavenay. *An Introduction to Machine Translation*. Thames and Hudson, London, 1960. Printed in Great Britain by The Camelot Press Ltd, Southampton.