Clinical Note Generation from Doctor-Patient Conversations using Decoder-Only Large Language Models

by

Saib Ahmed
22166032

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
M.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Saib Ahmed

22166032

Approval

The thesis/project titled "Clinical Note Generation from Doctor-Patient Conversations Using Decoder-Only Large Language Models" submitted by

1. Saib Ahmed (22166032)

Fall 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Sc. in Computer Science on October 28, 2024.

Examining Committee:

Examiner External:
(Member)

Dr. Rifat Shahriyar

Professor
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

Examiner Internal:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
BRAC University

Supervisor:
(Member)

Dr. Farig Yousuf Sadeque

Associate Professor
Department of Computer Science and Engineering
BRAC University

M.Sc. Coordinator:
(Member)

Dr. Md Sadek Ferdous

Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Dr. Sadia Hamid kazi

Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

I hereby declare that the contents of this thesis are entirely derived from our independent research. All external sources incorporated into this work have been duly acknowledged with full transparency. Furthermore, we affirm that neither this thesis nor any portion of it has been previously submitted or presented for the attainment of a degree at any other university or educational institution.

Abstract

Documenting clinical notes is a vital but time-consuming task in healthcare. Even in this modern era medical doctors spend considerable time documenting clinical notes from encounters with patients. While there have been significant advancements in general text summarization, research in clinical conversation summarization remains sparse due to the scarcity of open-source datasets available to the NLP community. Accurate summarization is paramount in clinical note generation, given its implications for human health. Our research demonstrates the efficacy of decoder-only models over traditional encoder-decoder models in generating more precise clinical notes from doctor-patient conversations. The study also tackles key challenges such as ensuring medical accuracy and complying with healthcare privacy standards. We utilized the **MTS-DIALOG** dataset [28], including $1,700$ such dialogues and corresponding clinical notes. This dataset was featured in the 2023 MEDIQAChat challenge, where the leading team, *WangLab* achieved a state-of-the-art (SOTA) Rouge-1 score of 0.4466 and BERTScore of 0.7307 [27]. Our study surpasses these benchmarks by fine-tuning the **"metallama/Meta-Llama-3-8B"** model enhanced with Qlora 8-bit quantization. We assessed our models using Rouge scores and BERT Scores to validate their superiority in performance. By evaluating the system on real-world clinical conversations, we show that the decoder-only LLM-generated notes closely match human-written ones in terms of completeness and clinical relevance. This research highlights the potential for decoder-only LLMs to revolutionize clinical workflows, making medical documentation more efficient while allowing doctors to focus more on patient care.

Keywords: ClinicalNLP; Dialouge2Note; Transformer; Decoder-Only; Mistral; Llama; Summarization; Rouge Score; BERTScore;

Dedication

I would like to remember my wonderful parents, without whom I would be worthless, with all of my sacrifices and academic endeavors.

Acknowledgement

Firstly, all praise to the Great Allah for whom my thesis has been completed without any major interruption.
Secondly, to my supervisor Dr. Farig Yousuf Sadeque sir for his kind support and advice in my work. He helped and advised me whenever I needed help.
And finally to my parents without their support, it may not be possible. With their kind support and prayer, I am now on the verge of completing my M.Sc.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Creating clinical notes manually has always been a time-consuming and exhausting job for healthcare providers. As healthcare systems grow more complex and larger in scale, it's become clear that there's a greater need for faster, more accurate ways to handle documentation. Since the invention of Transformer architecture, there have been significant advancements in several NLP tasks, including text summarization. Text summarization is an essential and common task in the field of NLP. Most of the recent progress is highly motivated by transformer-based large language models as well as the availability of large-scale datasets. The recent progress in this text summarization can improve the healthcare system. To enhance healthcare documentation and streamline the healthcare process, an NLP-powered system analyzes conversations, identifies relevant clinical facts, structures the information, and generates coherent medical reports. This can be done by automatically generating clinical notes by summarizing the conversation between doctor and patient. This ensures providing timely insights and support to medical professionals during patient interactions. While making important decisions, real-time information retrieval ensures that clinicians have access to relevant medical data and patient histories. This immediate support can lead to more accurate diagnoses, personalized treatment strategies, and improved patient outcomes. The ability to analyze data extends beyond individual patient interactions, allowing medical facilities to make data-driven decisions that enhance overall treatment quality, resource utilization, and patient satisfaction.

1.1 Motivation

Manual note-taking can be time-consuming, diverting healthcare providers' attention from patient care. On average medical doctors spend about 52 to 102 minutes daily writing clinical notes from their conversations with patients [3]. Automatic clinical note generation can be a solution to this problem. It can reduce the burden of paperwork on healthcare providers, and improve the accuracy of the medical records. This allows doctors to focus more on patient care rather than on paperwork. Moreover, during the COVID-19 pandemic, when in-person medical visits were getting limited, healthcare systems saw a greater than 100% increase in virtual urgent care visits and greater than 4000% increase in virtual non-urgent care visits [18]. Automatic clinical note generation can help us to overcome this kind of situation. Clinical notes can vary widely in terms of content, format, and quality. Automated systems can help to standardize documentation, improve data quality, and facilitate analysis. Nevertheless, Automated systems can extract valuable insights from clinical notes, enabling data-driven decision-making and improving patient care.

1.2 Problem Statement And Challenges

Generating clinical notes by summarizing doctor-patient conversations has its unique challenges. In this domain, accuracy is very important, as inaccuracies in critical medical facts can be extremely costly and may even endanger human lives. Hallucinations can also cause similar dire consequences. In this paper we addressed these challenges by employing decoder-only models rather than sequence-to-sequence models, thereby enhancing accuracy according to both ROUGE Score and BERT Score evaluation metrics compared to state-of-the-art summarization models.

For this task, we have used **MTS-DIALOG** dataset [28] which contains 1700 doctor-patient conversations (18k sentences) and their summarized clinical notes (6k sentences). (Abacha et al., 2023b) [28] studied several sequence-to-sequence summarization models and created a benchmark for the dataset. This dataset was also used in the 2023 MEDIQAChat challenge where the leading team surpassed the benchmark Rouge Scores and got a new state-of-the-art(SOTA) [27]. They have fine-tuned **FLAN-T5-Large** a Seq2Seq model, to get this performance.

The majority of seq2seq frameworks in use today have the Encoder-Decoder architecture [4], [5], where an encoder is responsible for encoding the input data into a hidden space, while a decoder is used to generate the target output text. In order to handle the Seq2Seq tasks, many promising large language models (**GPT** [9], **GPT2** [13], **GPT3** [17], **InstructGPT/ChatGPT** [25], **Palm** [30], **Bloom** [26], **Mistral** [31], **Llama** [40], **Llama2** [41], **Llama3** [46]) have emerged that use a language model that solely uses decoders. Despite the achievements of recent large language models, it is still not clear whether applying decoder-only models in the Seq2Seq task is a promising choice. According to (Liu et al., 2018) [8] the decoder-only language models get some gains over the Encoder-Decoder structure in the summarization task. In this paper, we mainly focused on the latest decoder-only models like **Mistral** [31] and **Llama** [40] and their several variants to generate clinical notes by summarizing the conversation between doctor and patient. We found Llama3 as our best performer which outperformed the best teams' performance at the 2023 MEDIQAChat challenge.

1.3 Research Objective

The core objective of this research is to explore, develop, and evaluate several decoder-only transformer models to generate summaries from doctor-patient conversations and use them as clinical notes. Our specific goals are:

- To identify a decoder-only transformer model suitable for clinical dialogue summarization that offers a balance between computational efficiency and high-quality summaries.

- To fine-tune the chosen transformer model on the **MTS-DIALOG** dataset, adapting its parameters to match the dataset's specific characteristics.
- To develop a state-of-the-art model that outperforms the current leading sequence-to-sequence model, Flan T5 Large, in generating clinical notes.
- To rigorously compare the ROUGE scores and BERT scores of the fine-tuned model with the current leading sequence-to-sequence model, **Flan-T5-Large**, to measure improvements in performance and summarization quality.

Chapter 2

Literature Review

The author of this paper describes an elaborate process for creating extensive decoder-only models to generate clinical notes from doctor-patient conversations. This task is approached as an abstractive text summarization problem. here several decoder-only models like the Llama and Mistral families were fine-tuned using parameter efficient fine-tuning (PEFT) [11] with low-rank adaptation (LoRA) [22].

2.1 Several Text Summarization LLMs

Model	Developed By	Model Architecture
T5	Google Research	Encoder-Decoder
FLAN-T5	Google Research	Encoder-Decoder
BART	FAIR	Encoder-Decoder
Mistral	Mistral AI	Decoder-Only
Llama	Meta AI	Decoder-Only

Table 2.1: Text Summarization LLMs

Generative AI transformer models have upended the status quo in Natural Language Processing and beyond. With their basis on the now-revolutionary Transformer Architecture, these models can generate human-quality text, translate languages, summarize texts, and write various creative content. In table 2.1 some popular LLMs are shown that are used for text summarization tasks.

2.1.1 Vanilla Transformer Model Overview

Transformer architecture is the backbone of almost all state-of-the-art (SOTA) LLMs. Transformers in NLP are a type of deep learning model that uses self-attention mechanisms to analyze and process natural language data. The Transformer architecture was introduced by Google researchers in the paper titled "Attention is all you need" [7]. The Transformer model uses an encoder-decoder structure, consisting solely of self-attention mechanisms and fully connected layers. A basic architecture of the model is shown in Figure 2.1. The architecture consists of:

- **Encoder**: A stack of N identical layers

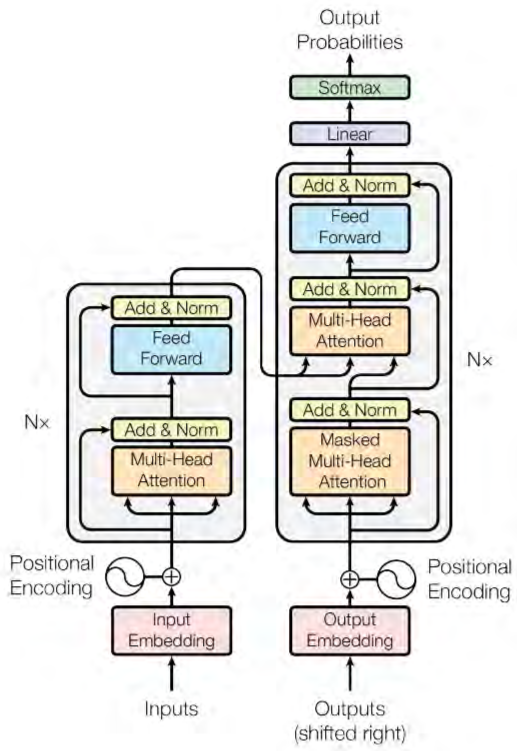

Figure 2.1: Illustration of the original transformer architecture proposed in Attention Is All You Need, 2017 [7]

- **Decoder**: Another stack of N identical layers.

Each of these layers is made up of two primary sub-layers:

- Multi-Head Self-Attention
- Position-wise Fully Connected Feed-Forward Network.

Self-Attention and Multi-Head Attention

Self-attention, also known as scaled dot-product attention, is a crucial mechanism in the transformer architecture that allows the model to weigh the importance of different tokens in the input sequence relative to each other. The key breakthrough of the Transformer model lies in its self-attention mechanism, which allows the model to weigh the importance of different tokens in a sentence relative to each other, regardless of their position. Self-Attention works as follows:

Input Representation:

For each word in the input sequence, the model constructs three vectors: **Query (Q)**, **Key (K)**, and **Value (V)**. These vectors are learned transformations of the input embeddings through learned weight matrices W_Q, W_K, and W_V.

Scaled Dot-Product Attention:

Self-attention is calculated using the dot product between the Query and Key vectors to determine the relevance of other words in the sequence. Mathematically, the attention score is calculated as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Where:

- QK^T computes the similarity between each query and key.
- The result is scaled by $\sqrt{d_k}$ to avoid large values in the dot product, which could push the softmax function into regions with very small gradients.
- The resulting weights are multiplied by the Value vectors, producing a weighted sum that determines the final representation for each word in the sequence.

Multi-Head Attention

The multi-head attention mechanism performs multiple self-attention calculations in parallel using different sets of Q, K, and V matrices, and then concatenates their outputs. Mathematically:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W_0 \tag{2.2}$$

Where, $head_i = Attention(QW_{Q_i}, KW_{k_i}, VW_{V_i})$
The concatenated outputs are then projected through an output weight matrix W_0.

Position-wise Feed-Forward Networks

Each encoder and decoder layer contains a fully connected feed-forward network that operates independently on each position in the sequence. This network consists of two linear transformations with a ReLU activation in between:

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b2 \tag{2.3}$$

Positional Encoding

Since the Transformer has no inherent sense of word order, positional encodings are added to the input embeddings to provide information about the relative positions of words in the sequence. These encodings are added element-wise to the input embeddings and are generated using sine and cosine functions of different frequencies:

$$PE(pos, 2i) = sin(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \tag{2.4}$$

$$PE(pos, 2i + 1) = cos(\frac{pos}{10000^{\frac{2i}{d_{model}}}}) \tag{2.5}$$

Encoder-Decoder Interaction

The encoder processes the input sequence and produces a sequence of continuous representations called context vectors. The decoder uses these context vectors along with the previously generated output (shifted right) to generate the next token in the output sequence. In the decoder, the multi-head attention mechanism is applied twice. The first one is similar to the encoder's self-attention but masked so that the decoder cannot attend to future tokens. This is important for autoregressive generation tasks. This kind of attention is known as Masked Multi-Head Attention. For the second one, the queries come from the previous decoder layer, while the keys and values come from the encoder's output. This layer allows the decoder to focus on relevant parts of the input sequence.

2.1.2 Text-to-Text Transformer (T5)

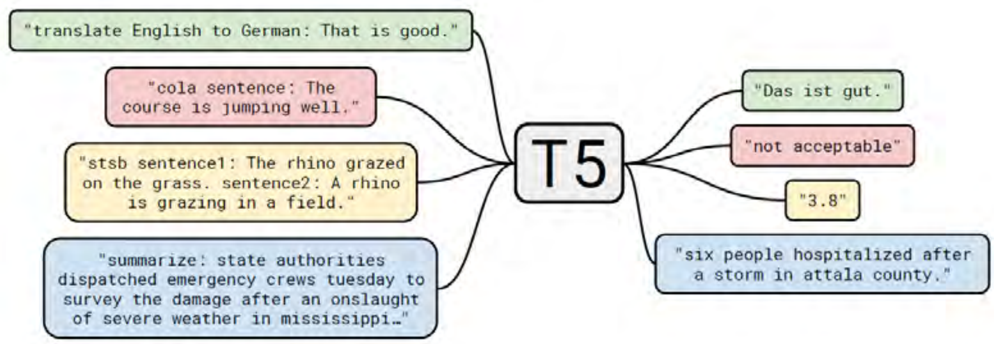

Figure 2.2: T5 model text-to-text framework [19]

Google Research developed the T5 model, which was proposed in the research paper [19]. It is designed to handle various NLP tasks using a unified "text-to-text" framework. The T5 model encodes the various tasks as text directives in the input stream, allowing it to handle a broad range of many-to-many and many-to-one NLP jobs uniformly. This makes it possible to train and supervise a single model for a broad range of natural language processing tasks, including summarization, translation, classification, Q&A, and even regression. A Figure of the framework is shown here 2.2. The T5 model is roughly equivalent to the original Transformer except for removing the Layer Norm bias, placing the layer normalization outside the residual path, and using a different position embedding scheme. T5 utilizes a training method known as "span corruption," in which random portions of the input text are hidden, and the model is tasked with generating the missing spans. This technique shares similarities with BERT's masked language model (MLM) training, where individual tokens are masked and predicted. The matrix representation of different attention masks is shown in Figure 2.3. However, T5 takes a more generalized approach by focusing on reconstructing longer sequences instead of single tokens, providing a broader and more flexible prediction capability. T5 is a highly versatile model that has become widely adopted due to its simple yet powerful text-to-text framework. Its encoder-decoder architecture and unified task formulation make it suitable for a wide variety of NLP tasks.

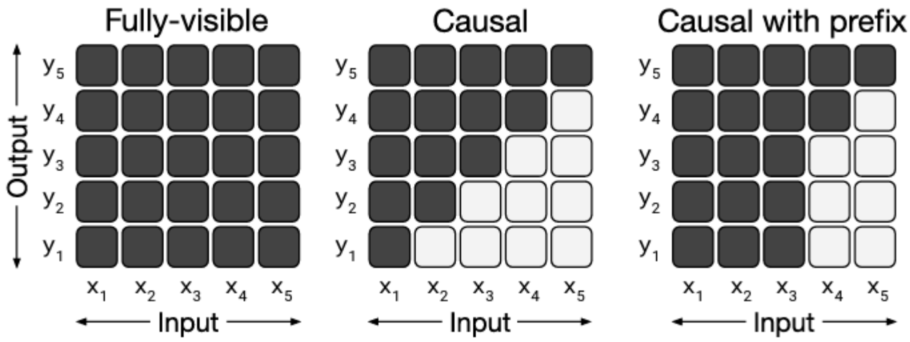

Figure 2.3: Matrices representing different attention mask patterns [19]

2.1.3 FLAN T5

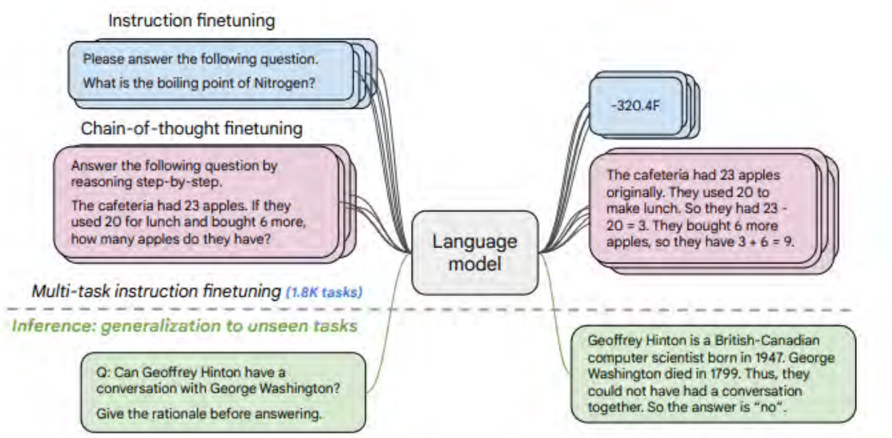

Figure 2.4: Fine-tuning of various language models [44]

FLAN-T5 preserves the core transformer-based encoder-decoder structure of T5. The bidirectional encoder captures contextual information from both directions in the input sequence, ensuring deep comprehension, while the autoregressive decoder generates text sequentially, from left to right, ensuring coherent language production. The model architecture was described in the paper [44]. The main aim was to improve the performance and generalization of language models through instruction fine-tuning, scaling model size, and incorporating chain-of-thought (CoT) data. The paper explores instruction fine-tuning with a particular focus on scaling the number of tasks, scaling the model size, and fine-tuning chain-of-thought data. The instruction fine-tuning is performed on a collection of data sources with a variety of instruction template types. This fine-tuning procedure is called Flan (fine-tuning language models) and a demonstration is shown in the figure 2.4. FLAN-T5's primary innovation lies in its ability to excel in few-shot learning, where the model is trained to perform well with minimal labeled examples. This is achieved through the integration of auxiliary networks, which provide additional context and guidance during learning. These networks help the model adapt quickly and accurately to new tasks, even when training data are scarce, significantly improving its generalization capabilities. Like T5, FLAN-T5 operates within a text-to-text framework, converting every NLP task into a text generation problem. This unified framework allows for easy fine-tuning, where the model can be adjusted using task-specific prompts. During fine-tuning, FLAN-T5 leverages its few-shot learning capabilities, making it highly efficient for tasks with limited datasets. This approach enhances its flexibility across diverse domains, improving its adaptability to new challenges.

2.1.4 BART

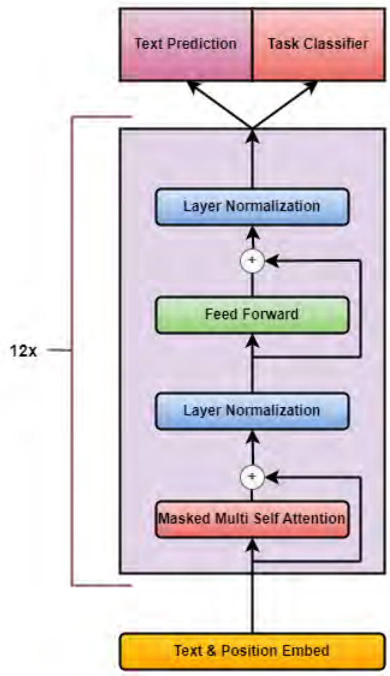

Figure 2.5: BART model architecture

BART (Bidirectional and Auto-Regressive Transformers) is a versatile sequence-to-sequence model developed by Facebook AI Research, optimized for a wide range of natural language processing (NLP) tasks. BART is a powerful language model introduced in 2019 in the paper [12]. It's designed for a wide range of natural language processing tasks, including text generation, summarization, and translation. BART's architecture is inspired by the Transformer model, which has proven to be highly effective for sequence-to-sequence tasks. BART large uses 12 layers in the encoder and decoder as shown in the figure 2.5. During training, BART uses a denoising autoencoder approach. The model is trained on text that has been corrupted by various noise functions (e.g., random deletions, permutations), and it learns to reconstruct the original text from these corrupted versions. This training strategy helps the model become proficient at recovering missing or noisy parts of the text and enhances its generative capabilities. In BART, the denoising autoencoder approach is applied to sequences of text. The training procedure involves two key steps:

- Corrupting the Input Text (Noising)
- Reconstructing the Original Text (Denoising)

BART's use of the denoising autoencoder approach allows it to be bidirectional, like BERT, while also being auto-regressive, like GPT. This enables BART to be effective at both Understanding tasks and Generation tasks. BART's encoder operates similarly to BERT, processing the entire input sequence and leveraging bidirectional attention to understand the relationships between tokens. The decoder functions more like GPT, generating output sequences in an auto-regressive manner, meaning it predicts each token one by one, conditioned on the tokens generated before it. By combining these elements, BART's training process effectively teaches the model to reconstruct meaningful text from a corrupted input, giving it a strong performance in both text understanding and generation.

2.1.5 Mistral

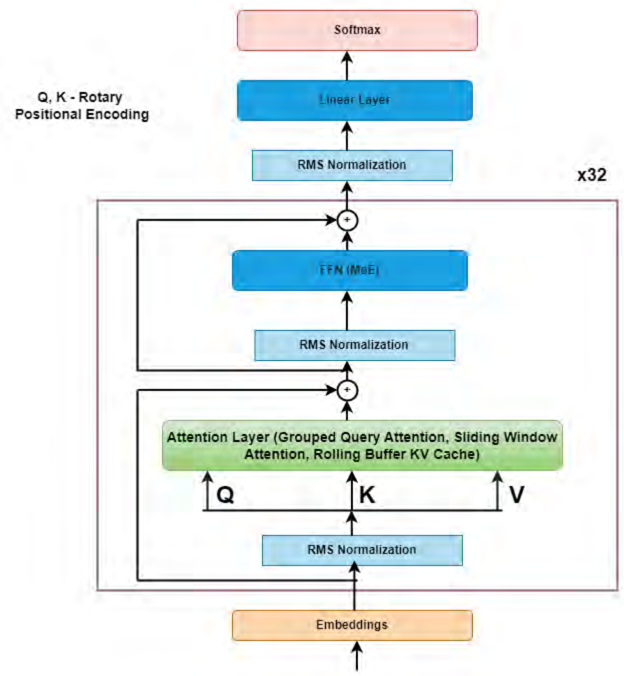

Figure 2.6: Mistral 7 X 8B Architecture

The Mistral model's architecture was introduced in the paper [31]. The Mistral model is a state-of-the-art decoder-only large language model (LLM) that redefines the landscape of natural language processing (NLP). Mistral does not use a traditional encoder-decoder structure but instead follows the structure of a decoder-only model. This follows the pattern of modern language models like GPT, where text generation is the focus, and self-attention is applied across all tokens. The Mistral model's architecture is shown in figure 2.6. Another key feature of the Mistral model is its parameter efficiency. Despite having fewer parameters than many large-scale models, it achieves a high level of accuracy and deep contextual understanding. This efficiency is a product of advanced optimization techniques, which allow Mistral to deliver robust performance even in resource-constrained environments. This makes the model highly suitable for deployment on edge devices or in settings where low latency is crucial. The model is trained on vast datasets and requires substantial computational power for fine-tuning. Through this process, Mistral can perform exceptionally well across various NLP tasks, including text generation, machine translation, and text comprehension. Its fine-tuning enhances the model's ability to generate coherent and contextually appropriate responses in a wide range of applications.

Sliding Window Attention

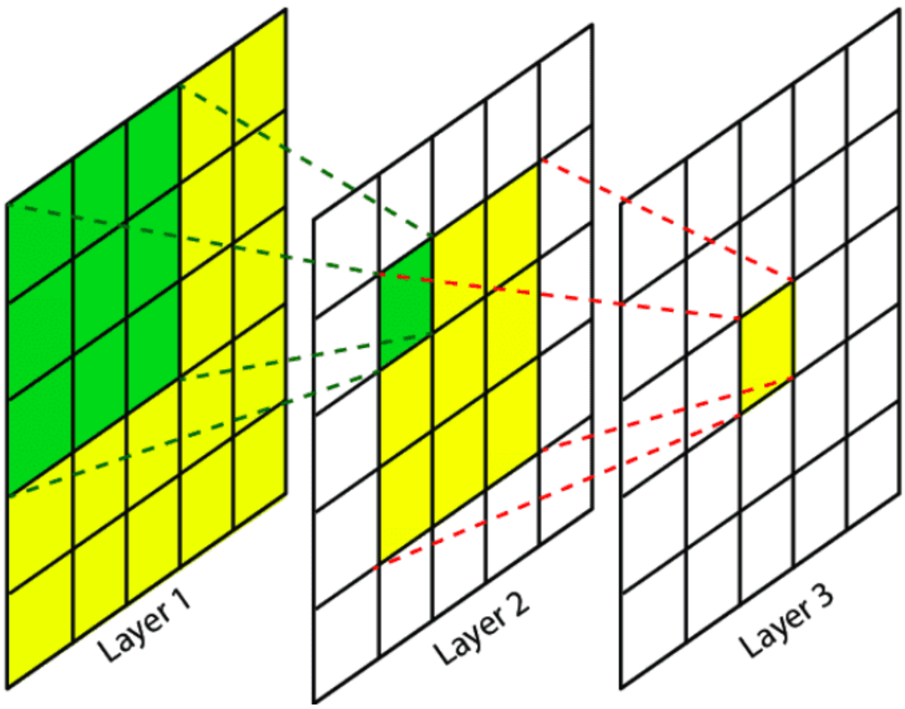

Figure 2.7: The receptive field of each convolution layer with a 3×3 kernel [6]

Sliding window attention is a technique used in models like Mistral to handle long sequences more efficiently. In large Transformer models, standard self-attention has quadratic complexity, which becomes computationally expensive when dealing with long inputs. Sliding window attention reduces this overhead by limiting the attention computation to local neighborhoods of tokens rather than the entire sequence. It reduces the number of dot-products to perform, and thus, performance during training and inference. Sliding window attention may lead to degradation in the performance of the model, as some "interactions" between tokens will not be captured. The model mostly focuses on the local context, which depending on the size of the window, is enough for most cases. Sliding window attention can still allow one token to watch tokens outside the window, using reasoning similar to the receptive field in convolutional neural networks shown in figure 2.7. Figure 2.8 illustrates the architectural details of sliding window attention in mistral. From these figures, it is also visible that the information flow is quite similar to the receptive field of a CNN.

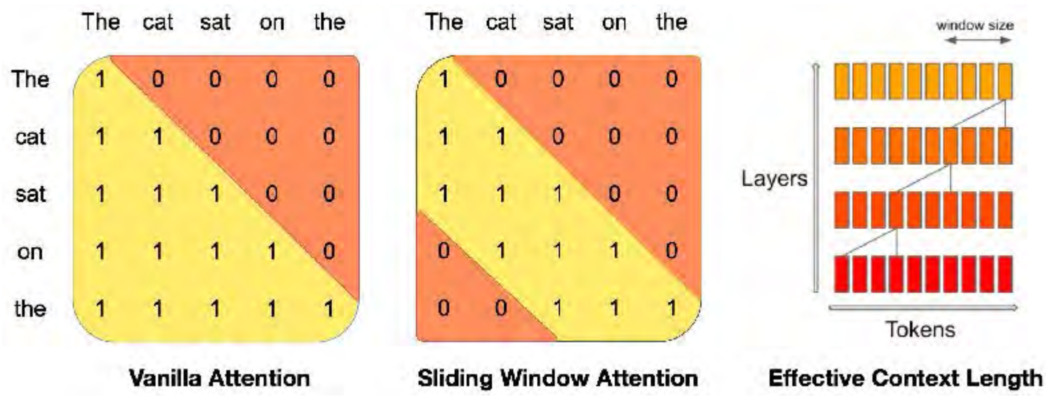

Figure 2.8: Architectural details of sliding window attention [31]

KV cache

KV Cache (Key-Value Cache) is a type of data store that efficiently stores and retrieves small pieces of data, typically referred to as "key-value pairs". It's optimized for high-speed access and is often used to improve the performance of applications that require frequent lookups of frequently accessed data. In transformer models, self-attention computes the relationship between each token (word) in a sequence. For every token, it calculates the Query (Q), Key (K), and Value (V) matrices, which help determine how much focus each token should have on other tokens in the sequence [7]. When generating sequences token by token, the same computations (K and V matrices) for past tokens are reused at each new step. Instead of recalculating these matrices every time a new token is generated, the KV cache stores them for all previously processed tokens. An illustration is shown in the figure 2.9 This prevents redundant calculations, making the generation process faster and more efficient.

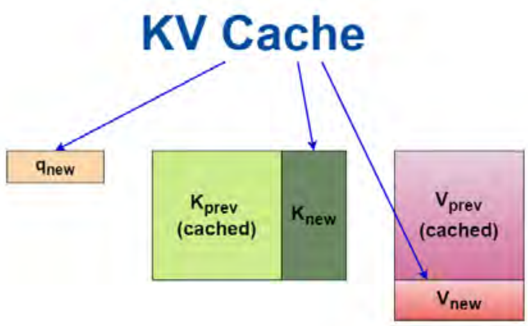

Figure 2.9: KV Cache

Rolling Buffer Cache

The rolling buffer cache in Mistral is a memory-efficient mechanism used to handle long-context sequence generation. It's particularly useful when the model needs to process large inputs or generate long outputs without running into memory constraints or significant computational slowdowns. In autoregressive text generation, transformers typically require a full history of previous tokens to calculate attention for the current token. The rolling buffer cache optimizes this by limiting how much of the past sequence is retained in memory, while still allowing the model to generate text based on recent context. Transformer models have a maximum context length, beyond which they cannot attend to tokens effectively. The rolling buffer cache manages this by maintaining a fixed-size buffer, which stores the most recent tokens. As the model generates new tokens, the buffer "rolls" forward, discarding the oldest tokens and retaining only the most recent ones. This sliding window ensures that the model always has access to the most relevant recent context while keeping memory usage efficient. The rolling buffer cache works alongside the KV cache. Instead of keeping K and V matrices for all tokens ever processed, the rolling buffer keeps these matrices only for the tokens within the current context window. When the window moves forward, the older K and V matrices are dropped to save memory and computation time.

Mixture of Experts (MoE)

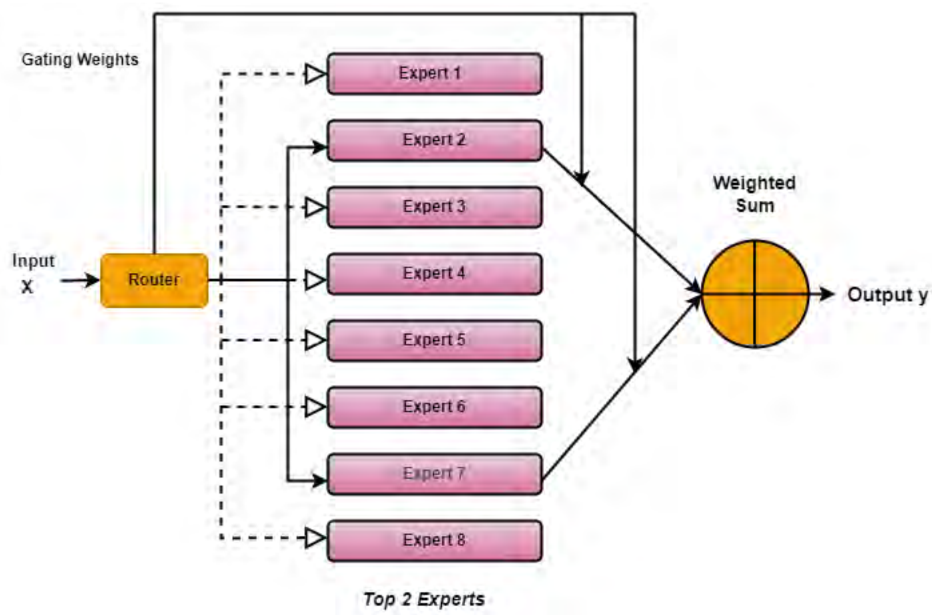

Figure 2.10: Sparse Mixture of Experts (SMoE)

In Mistral, a Mixture of Experts (MoE) is indeed a sophisticated ensemble technique used to improve model efficiency and performance. It is designed to allocate computational resources dynamically by activating only a subset of "expert" neural networks (or submodels) for any given input, rather than using the entire model all the time. This makes the model more computationally efficient, especially when scaling to larger architectures. The Mixture of Experts (MoE) is an ensemble technique that combines several expert models, much like how the Random Forest algorithm brings together multiple decision tree models. While Random Forest consists of decision trees, MoE is made up of feedforward neural networks (FFNs). In MoE, each expert focuses on a specific aspect of the task or input data, allowing them to specialize in handling particular problem segments. This specialization enables each model to excel in its designated tasks, improving overall performance. In the case of Mistral 8x7B, a Sparse Mixture of Experts (SMoE) is discussed [24]. In a Sparse MoE, each token is only sent to the top k experts, rather than all the experts in the MoE layer. This means that only a selected few experts are activated for each token, rather than all of the experts. In the Mistral model, the router is responsible for selecting the top 2 experts for each token. An illustration of SMoE is shown in Figure 2.6 This approach reduces the computational cost and makes the model more efficient, while still allowing the experts to specialize and learn effectively.

2.1.6 Llama

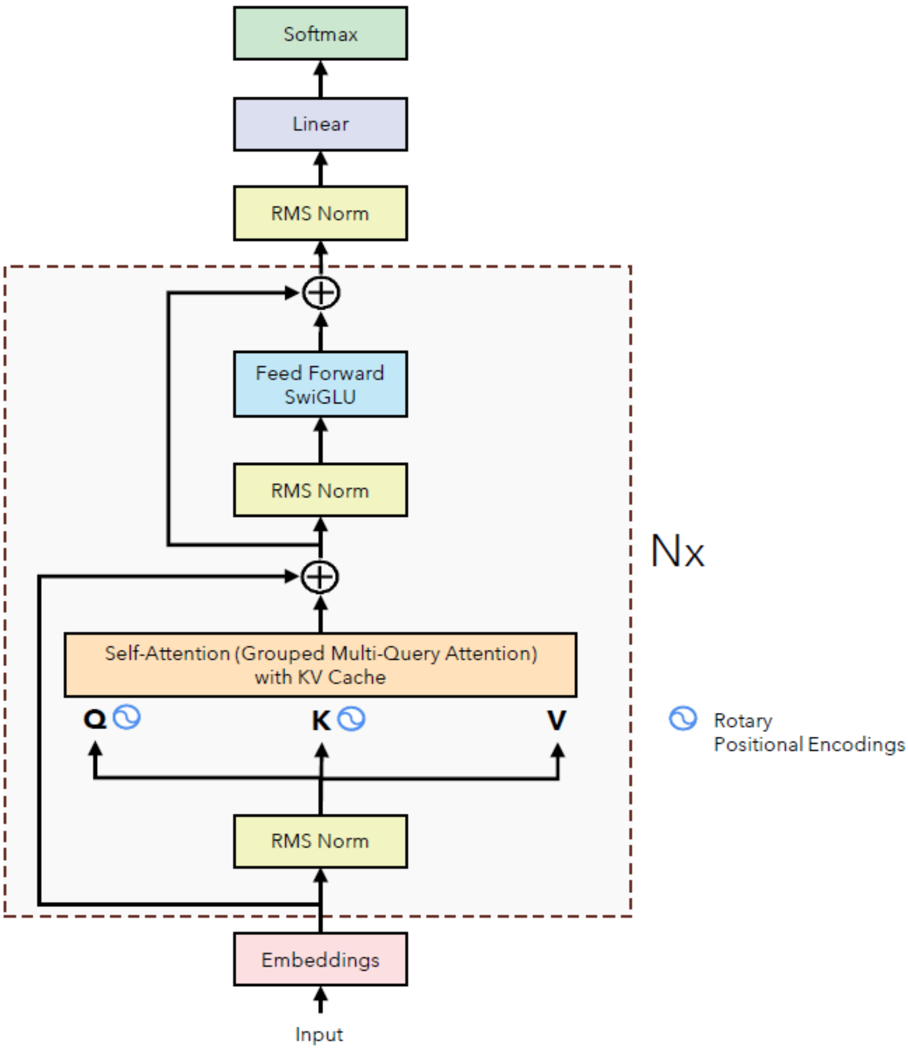

Figure 2.11: Llama architecture

Llama is a groundbreaking series of open-source large language models developed by Meta. Llama 1 was initially introduced in four different versions, featuring parameter sizes of 6.7B, 13B, 32.5B, and 65.2B. Each version's multi-head attention system contains 32, 40, 52, and 64 heads, respectively, unlike the transformer model, which had only 8 heads in its multi-head attention. It's worth noting that Llama employs a slightly adjusted form of multi-head attention, known as grouped multi-query attention. In Llama, every token within the input embedding is represented by vectors of various dimensions based on the model's size. Specifically, the 6B parameter model uses a 4096-dimensional vector for each token, while the 13B model increases the dimension to 5120. For the 32.5B model, the dimensionality grows to 6656, and for the 65.2B model, tokens are represented by 8192-dimensional vectors. This contrasts with the original transformer model, where each token is represented by a 512-dimensional vector. Additionally, the input embeddings in Llama models are dynamic and are learned during the training process. Meta launched Llama 3, the latest in its Llama series of open-source AI models. Llama 3 comes in two variants: one with 8 billion parameters and another with 70 billion parameters. The key difference between the predecessor's models is, the size of the pretraining corpus increased by 650% LLaMA — 2 was trained on 2T tokens whereas LLaMA — 3 was trained on 15T tokens, doubled the context length of the model from 4K to 8K on both 8B and 70B models, and adopted grouped-query attention for both 8B and 70B variant as compared to the previous generation (GQA) was only used in bigger models 34B and 70B. LLaMA 3 imbibes its Architecture from its previous generation models [40], [41], [46]. The basic architecture of the Llama model is shown in the figure 2.11.

RoPE (Rotary Positional Encoding)

One of the key differences between the Llama and the vanilla transformer model is Llama uses Rotary Positional Encoding (RoPE). Absolute positional encodings are predefined vectors added to a token's embedding to indicate its exact position within a sentence. This method processes each token individually. Relative positional encodings, however, operate on pairs of tokens and are utilized during attention calculations. Since the attention mechanism measures the degree of relatedness between two words, relative positional encodings provide information about the distance between those two words to the attention mechanism. So a vector was created of two given tokens that represent their distance. Relative positional encodings were introduced in the paper [10].

Rotoary Positional Encoding can be considered as a midground between Absolute Positional Embeddings and Relative Positional Embeddings as each token does have a fixed or an absolute embedding value and is multiplied by an inner dot product with its polar form which is relative to the rotation of the vectors on the 2D plane. Rotoary Positional Encoding was introduced in the paper [47]. The dot product used in the attention mechanism is a type of inner product, which can be thought of as a generalization of the dot product. An inner product can be found over the two vectors q (query) and k (key) used in the attention mechanism, which depends only on the two vectors and the relative distance of the tokens they represent. In the paper a function g like the following that only depends on the two embeddings vector q and k and their relative distance was defined.

$$f_q(x_m, m) = (W_q x_m)e^{im\theta} \tag{2.6}$$

$$f_k(x_n, n) = (W_k x_n)ein\theta \tag{2.7}$$

$$g(x_m, x_n, m-n) = Re[(W_q x_m)(W_k x_n) * e^{i(m-n)\theta)}] \tag{2.8}$$

After using Euler's formula, its matrix can be written as:

$$f_{\{q,k\}}(x_m, m) = \begin{pmatrix} \cos m\theta & -\sin m\theta \\ \sin m\theta & \cos m\theta \end{pmatrix} \begin{pmatrix} W^{(11)}_{\{q,k\}} & W^{(12)}_{\{q,k\}} \\ W^{(21)}_{\{q,k\}} & W^{(22)}_{\{q,k\}} \end{pmatrix} \begin{pmatrix} x^{(1)}_m \\ x^{(2)}_m \end{pmatrix} \tag{2.9}$$

In equation 2.9 a rotation matrix that is the rotation of some vector in the 2D space and the rotation is dependent on m and θ where m is the absolute position of the token and θ is the rotation angle. A visualization is shown in Figure 2.12.

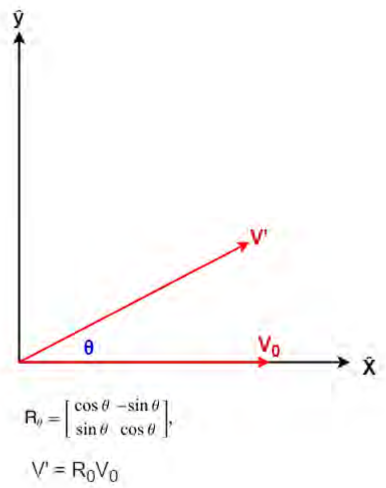

Figure 2.12: Rotational Matrix

KV Cache

The KV cache in Llama is similar to that described in Mistral in section 2.1.5. At every step of the inference, only the last token output by the model is of interest, as the previous ones are already known. However, to determine the next token to output, the model requires access to all the preceding tokens, which serve as its context. It's a way to make the model do less computation on the token it has already seen during inference. It was made possible by using the KV Cache technique.

Grouped Multi-Query Attention

Multi-head attention layers, used in the Transformer neural sequence model, offer a powerful alternative to RNNs for transferring information within and across sequences [7]. Training these layers is typically fast and straightforward due to their

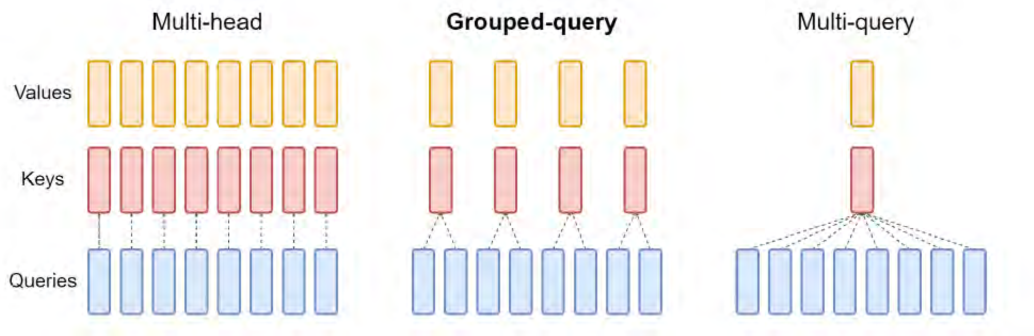

Figure 2.13: Overview of Multi-Head, Grouped Multi-Query and Multi-Query attention [29]

ability to be parallelized across the sequence length. However, incremental inference, where such parallelism isn't possible, tends to be slower because of the high memory-bandwidth cost associated with repeatedly loading the large "keys" and "values" tensors. To address this, we introduce a variation called multi-query attention, where the keys and values are shared among all attention heads, significantly reducing the tensor size and memory bandwidth needed for incremental decoding. Experimental results confirm that this approach speeds up decoding and results in only minimal quality loss compared to the original model. Multi-query attention is a mechanism in transformer models that allows a single query to attend to multiple key-value pairs simultaneously. This is different from the standard self-attention mechanism, where each query attends to all key-value pairs in the sequence [14].

Grouped-query attention (GQA) was introduced in the paper [29]. Grouped-query attention (GQA) combines elements of multi-query and multi-head attention, delivering quality comparable to multi-head attention while offering speeds similar to multi-query attention. In autoregressive decoding, it's common to cache the keys and values of prior tokens in a sequence to expedite attention computation. However, as the context window or batch size grows, the memory cost of storing the key-value cache (*kv* cache) in multi-head attention (MHA) models increases significantly. Multi-query attention (MQA) addresses this by using a single key-value head for multiple queries, reducing memory use and speeding up decoder inference. An overview of Multi-Head, Grouped Multi-query, and Multi-Query attention illustration is shown in figure 2.13. Llama employs GQA to mitigate memory bandwidth limitations during the autoregressive decoding of Transformer models. The main challenge arises from the GPU performing computations faster than it can transfer data into memory, compounded by the large memory required to load decoder weights and attention keys at each step.

Root Mean Squared Normalization

Root Mean Square Normalization (RMSNorm) is a relatively novel normalization technique introduced by Biao Zhang and Rico Sennrich in 2019 [15]. Unlike Batch Normalization and Layer Normalization, Root Mean Squared Normalization normalizes activations based on the root mean square of the activations themselves, rather than using mini-batch or layer statistics. This approach ensures that the activations are consistently scaled regardless of the mini-batch size or the number of features. Additionally, RMSNorm introduces learnable scale parameters, offering similar adaptability to Batch Normalization. RMSNorm only focuses on re-scaling invariance and regularizes the summed inputs simply according to the root mean square (RMS) statistic:

$$\bar{a_i} = \frac{a_i}{\mathrm{RMS}(a)} g_i, \quad \text{where} \quad \mathrm{RMS}(a) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_i^2} \tag{2.10}$$

SwiGLU Activation Function

SwiGLU is an activation function used in deep neural networks that is a variant of GLU (Gated Linear Unit) and was introduced in the paper [20]. SwiGLU is used to compute the output of a neuron in a neural network by taking the weighted sum of the inputs and applying a non-linear function to it. Mathematically, SwiGLU involves the Swish function and tensor multiplication. As a variant of GLU, SwiGLU is built on the same fundamental mathematical principles. However, it differs by employing the Swish function as its non-linear component. The Swish function, a newer activation function, has been shown to outperform other activation functions in certain use cases. SwiGLU offers several advantages, making it an effective activation function for neural networks. First, it builds on the GLU concept, demonstrating strong performance across various applications. Second, SwiGLU utilizes the Swish function, which has been found to outperform other activation functions, especially when used alongside residual connections. Lastly, it supports efficient computation due to its reliance on element-wise multiplication. Mathematical expression of the function is:

$$\mathrm{FFN}_{\mathrm{SwiGLU}}(x, W, V, W_2) = (\mathrm{Swish}_1(xW) \otimes (xV))W_2 \tag{2.11}$$

The GLU family of layers has been extended and proposed for use in Transformer. In a transfer-learning setup, the new variants appear to generate better perplexities for the de-noising objective employed in pre-training, as well as improved results on numerous downstream language-understanding tasks. These architectures are straightforward to implement and exhibit no discernible computational disadvantages. No explanation is provided for the effectiveness of these architectures; their success, like everything else, is attributed to divine benevolence.

2.2 LLMs in healthcare industry

In "An Empirical Study of Clinical Note Generation from Doctor-Patient Encounters," [28] Asma Ben Abacha and colleagues present the MTS-DIALOG dataset, a substantial compilation of simulated doctor-patient dialogues linked to corresponding clinical notes. The primary aim of the study was to assess the viability of automatically generating clinical notes through advanced transformer models, such as BART and Pegasus. The researchers evaluated these models using both automated metrics—such as ROUGE, Fact Scores, and BLEURT and manual assessments conducted by experts. The findings indicate that BART, particularly when pre-finetuned and applied with guided summarization techniques, delivered the highest accuracy in generating clinical notes. Nonetheless, persistent challenges, including hallucinations and the omission of key facts, remain. This study not only offers a foundational dataset and key performance benchmarks but also underscores the potential for reducing the documentation workload in healthcare. At the same time, it highlights the need for ongoing improvements, especially in enhancing factual accuracy and better capturing the nuances of real-world clinical conversations.

The authors of this paper [34] investigate the application of large language models (LLMs) for summarizing and classifying medical dialogues. As part of the MEDIQA-Chat 2023 competition, the researchers sought to generate section summaries and headers from conversations between patients and doctors. They experimented with several models, including T5-Small, T5-Base, and Clinical-T5 for summarization, and Roberta-base for classification tasks.
Using the MIMIC datasets, the team fine-tuned these models and applied data augmentation techniques to boost performance. Their highest accuracy, 72.3%, was achieved with the Clinical-T5-Sci model for summarization and Roberta-base for header classification. Despite these successes, they encountered challenges related to limited hardware and dataset size. The study highlights the promise of LLMs in automating medical document generation, though future research should explore more advanced models and improved pre- and post-processing techniques for even better outcomes.

In this paper [36], authors present their approach for the MEDIQA-Chat 2023 competition, focusing on the use of large language models (LLMs) to generate, augment, and summarize patient-doctor conversations. They employed BART-large, fine-tuning it on datasets such as SAMSum and MIMIC-IV-Note. To address the challenge of summarizing lengthy medical dialogues, they introduced a novel N-pass strategy, where conversation blocks are processed sequentially alongside partial summaries. Data augmentation played a crucial role in improving performance, particularly through the use of synthetic dialogues derived from MIMIC-IV-Note. Despite achieving strong results, the models faced challenges such as hallucinations and the omission of critical medical concepts. Future directions include incorporating external medical knowledge and enhancing dialogue-generation techniques. This research contributes to advancing clinical natural language processing (NLP) tasks and automating medical documentation processes.

The authors of this paper [42] investigated the use of large language models (LLMs) to generate synthetic doctor-patient dialogues for the MEDIQA-Chat 2023 competition. They introduced a novel doctor-patient loop system that utilized ChatGPT and BioMedLM to create clinically relevant conversations. The team applied their approach to tasks such as clinical note summarization and dialogue generation, achieving notable success. BioMedLM performed well in classifying note sections, while ChatGPT excelled in generating realistic dialogue content. Their findings demonstrate that when fine-tuned and properly segmented, LLMs can effectively simulate doctor-patient interactions. However, challenges remained, such as gaps in medical knowledge and difficulties in handling lengthy conversations. Future research will focus on incorporating more specialized medical knowledge and improving dialogue segmentation.

Authors of this paper [38] conducted a comprehensive evaluation of various Transformer-based models, including BioBart, Flan-T5, DialogLED, and OpenAI's GPT-3, in the context of the MEDIQA-Chat 2023 challenge, which focused on summarizing clinical dialogues. Their aim was to automate the process of generating clinical note sections from doctor-patient conversations. The study employed a variety of techniques, such as fine-tuning, ensemble learning, and GPT-3's in-context learning, to create accurate summaries. Among these, DialogLED-Large demonstrated superior performance, especially in handling lengthy conversations, surpassing GPT-3, which showed vulnerabilities to generating hallucinations. Notably, the research illustrated that well-tuned models could match the efficiency of GPT-3 while offering a more cost-effective and secure solution for healthcare applications. However, a major constraint was the limited training data, which hindered overall performance. To enhance model reliability and adaptability for real-world clinical use, the authors suggest future exploration of data augmentation methods and strategies to mitigate hallucinations.

Authors of this paper [39] took part in the MEDIQA-Chat 2023 challenge, aiming to generate clinical notes from doctor-patient interactions. They fine-tuned models like BART, RoBERTa, and CONFIT, while also leveraging GPT-4 for in-context learning to enhance clinical note generation. Their approach yielded competitive outcomes, with the CONFIT model using a dynamic max-length strategy and GPT-4 excelling in full note generation. Although automated metrics like ROUGE and BERTScore were employed for evaluation, human experts favored GPT-4's outputs due to their superior accuracy and more natural, human-like quality. However, concerns were raised about patient privacy when using external APIs for medical data processing. The authors emphasized the need for more robust automated evaluation methods and underscored the effectiveness of large language models in handling the complexities of clinical dialogues. These findings demonstrate the promising role that advanced models like GPT-4 could play in streamlining healthcare documentation.

In this paper [37] authors investigated transformer-based ensembling techniques to enhance clinical conversation summarization as part of the MEDIQA-Chat 2023 challenge. His research evaluated three distinct methods using the LSG BART model: a single model (both with and without fine-tuning on PubMed data), a section-wise ensemble model, and a multi-layer summarization approach. The most promising results emerged from the section-wise ensemble models, where specialized models were assigned to summarize different sections of chart notes, yielding greater accuracy compared to single-model approaches. On the other hand, multi-layer techniques failed to improve performance, and fine-tuning on PubMed data—more focused on medical literature rather than dialogue—actually decreased accuracy. The findings indicate that model specialization for distinct sections of clinical reports holds significant potential, though further exploration is necessary to refine multi-layer models and identify more suitable data sources for fine-tuning.

Authors of this paper [33] explored advanced techniques for improving clinical note generation as part of the MEDIQA-Chat 2023 shared task. They fine-tuned two T5-based models, FLAN-T5 and LongT5, using datasets of doctor-patient conversations, focusing on multi-task learning to enhance the quality of medical summaries while minimizing hallucinations. Their methodology centered on text-to-text modeling, with experiments involving different text generation strategies such as beam search and contrastive search. The results demonstrated that multi-task fine-tuning significantly enhanced performance, particularly by reducing factual inaccuracies. However, incorporating clinical named entity recognition (NER) tags as part of data augmentation unexpectedly worsened the quality of the generated summaries. These findings suggest that while multi-task learning is a promising avenue for improving medical note generation, further optimization of NER-based data augmentation is needed to avoid negatively impacting summarization outcomes.

In this paper [43], the authors introduced an innovative hybrid approach for medical dialogue summarization during the MEDIQA-Chat 2023 challenge. Their method integrated a Support Vector Machine (SVM) for dialogue classification with GPT-3 models for summarization. They executed two runs: one using GPT-3.5 with one-shot prompts and the other utilizing a fine-tuned GPT-3 Curie model. The results showed that the GPT-3.5 model outperformed the GPT-3 Curie variant, achieving superior metrics across ROUGE-1, BERTScore, and BLEURT evaluations. Additionally, the SVM classifier achieved a 70% accuracy rate, exceeding the average performance of other participants. This research underscores the potential of combining traditional machine learning techniques with state-of-the-art language models to enhance medical dialogue summarization. However, further refinement of both the classification accuracy and summarization prompts is needed. Overall, the study contributes valuable insights, showcasing the promise of hybrid models in assisting healthcare professionals in managing extensive clinical information.

In the paper titled ”SummQA at MEDIQA-Chat 2023: In-Context Learning with GPT-4 for Medical Summarization,” [32] authors introduce a novel system for summarizing medical dialogues. Their methodology leverages GPT-4’s capabilities with in-context examples to generate clinical summaries from doctor-patient interactions. For section-specific summaries, the team selected dialogues with semantic similarity, while full-note summarization relied on a single example due to constraints on input length. The system performed exceptionally well, securing top rankings in the MEDIQA 2023 Shared Task. Despite GPT-4’s ability to generate concise, abstractive summaries, the authors identified challenges, including the tendency to produce overly brief summaries and the risks of privacy concerns when handling real patient data. This work represents a significant step forward in applying large language models to medical summarization, particularly in addressing data scarcity in critical healthcare contexts.

Chapter 3

Description of MTS-DIALOG Corpus

3.1 Main Dataset

```
Dialogue:

('Doctor: What brings you back into the clinic today, miss? \r\n'
 'Patient: I came in for a refill of my blood pressure medicine. \r\n'
 'Doctor: It looks like Doctor Kumar followed up with you last time regarding '
 'your hypertension, osteoarthritis, osteoporosis, hypothyroidism, allergic '
 'rhinitis and kidney stones.  Have you noticed any changes or do you have any '
 'concerns regarding these issues?  \r\n'
 'Patient: No. \r\n'
 'Doctor: Have you had any fever or chills, cough, congestion, nausea, '
 'vomiting, chest pain, chest pressure?\r\n'
 'Patient: No.  \r\n'
 'Doctor: Great. Also, for our records, how old are you and what race do you '
 'identify yourself as?\r\n'
 'Patient: I am seventy six years old and identify as a white female.')

Notes:

('The patient is a 76-year-old white female who presents to the clinic today '
 'originally for hypertension and a med check.  She has a history of '
 'hypertension, osteoarthritis, osteoporosis, hypothyroidism, allergic '
 'rhinitis and kidney stones.  Since her last visit she has been followed by '
 'Dr. Kumar.  Those issues are stable.  She has had no fever or chills, cough, '
 'congestion, nausea, vomiting, chest pain, chest pressure.')
```

Figure 3.1: Example of data point of MTS-DIALOG Dataset, Dialogue, and Notes

The MTS-Dialog dataset is a new collection of 1.7k short doctor-patient conversations and corresponding summaries (section headers and contents) [28].

- The training set consists of 1,201 pairs of conversations and associated summaries.
- The validation set consists of 100 pairs of conversations and their summaries.

- MTS-Dialog includes 2 test sets; each test set consists of 200 conversations and associated section headers and contents:
  - MTS-Dialog-TestSet-1-MEDIQA-Chat-2023.csv: Official test set used in the MEDIQA-Chat 2023 challenge (Task A)
  - MTS-Dialog-TestSet-2-MEDIQA-Sum-2023.csv: Official test set used in the MEDIQA-Sum 2023 challenge (Task A & Task B)

A sample data point is shown in the figure 3.1

3.1.1 Section-header categories

The MTS-Dialog dataset is divided into 20 categories of section headers. The distribution of section headers is shown in figure 3.2 and the statistics of the dataset shared in the paper [28] can be found in the table 3.2. The full list of normalized section headers and their counts are shown in the table 3.1

Section Headers	**Counts**
FAM/SOCHX [FAMILY HISTORY/SOCIAL HISTORY]	465
GENHX [HISTORY of PRESENT ILLNESS]	392
PASTMEDICALHX [PAST MEDICAL HISTORY]	168
CC [CHIEF COMPLAINT]	105
PASTSURGICAL [PAST SURGICAL HISTORY]	86
ALLERGY	84
ROS [REVIEW OF SYSTEMS]	98
MEDICATIONS	80
ASSESSMENT	59
EXAM	34
DIAGNOSIS	23
DISPOSITION	22
PLAN	22
EDCOURSE [EMERGENCY DEPARTMENT COURSE]	16
IMMUNIZATIONS	11
IMAGING	10
GYNHX [GYNECOLOGIC HISTORY]	8
PROCEDURES	6
OTHER_HISTORY	7
LABS	5

Table 3.1: Section Headers in the Dataset.

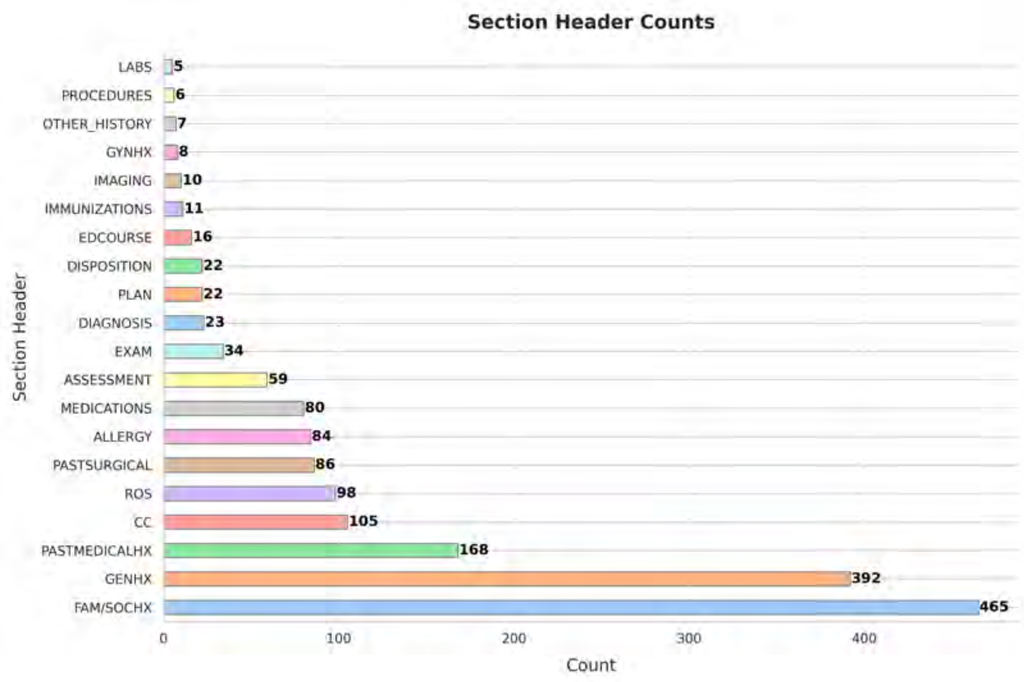

Figure 3.2: Section Header distribution of MTS-Dialog dataset

	Dialogue			Summary	
	Turns	**Sentences**	**Words**	**Sentences**	**Words**
count	15,969	18,406	241,685	5,870	81,299
mean	9	11	142	3	48
max	103	136	1,951	57	1,182
25-perc	4	4	48	1	6
50-perc	6	7	88	2	18
75-perc	12	14	176	4	55

Table 3.2: Statistics of the MTS-DIALOG Dataset.

3.1.2 Data Quality

The MTS-DIALOG dataset undergoes a thorough three-step process to ensure its quality. First, only those with medical backgrounds, such as former medical scribes, were selected to serve as annotators. Second, during the early stages of their work, each annotator received one-on-one feedback from an experienced trainer to help refine their skills. Lastly, after the dataset was completed, an independent validation process took place. This separate review used a grading rubric to assess how well the annotated conversations followed the guidelines and how relevant the content was to the original clinical notes. Minor corrections, like fixing typos or filling in missing information, were made during this stage to make sure the final dataset was even more accurate than the initial version [27]. Even though there was one annotation error in the dataset that was shown in the figure 6.1.

3.1.3 Comparison with Real Data

The MTS-DIALOG dataset includes both real medical notes and simulated conversations that mirror doctor-patient interactions, helping to avoid any breaches of confidentiality. To understand the impact of relying heavily on synthetic data, a blind review was conducted to compare the MTS-DIALOG data with real conversations. Distinguishing between the simulated and real data in the dataset is a challenging task. While statistical analysis shows that the simulated conversations have fewer speech errors and pauses, medical experts noted that the dialogues generally feel authentic. In some cases, the clarity, directness, and ease of understanding—even with sudden shifts in topics—led to synthetic data being mistaken for real interactions. On the other hand, actual data, known for its honesty and minimal speech flaws, was often confused for simulated content due to its polished nature. This difficulty highlights the dataset's value as a foundation for training and evaluating models in practical, real-world settings.

3.2 Augmented dataset

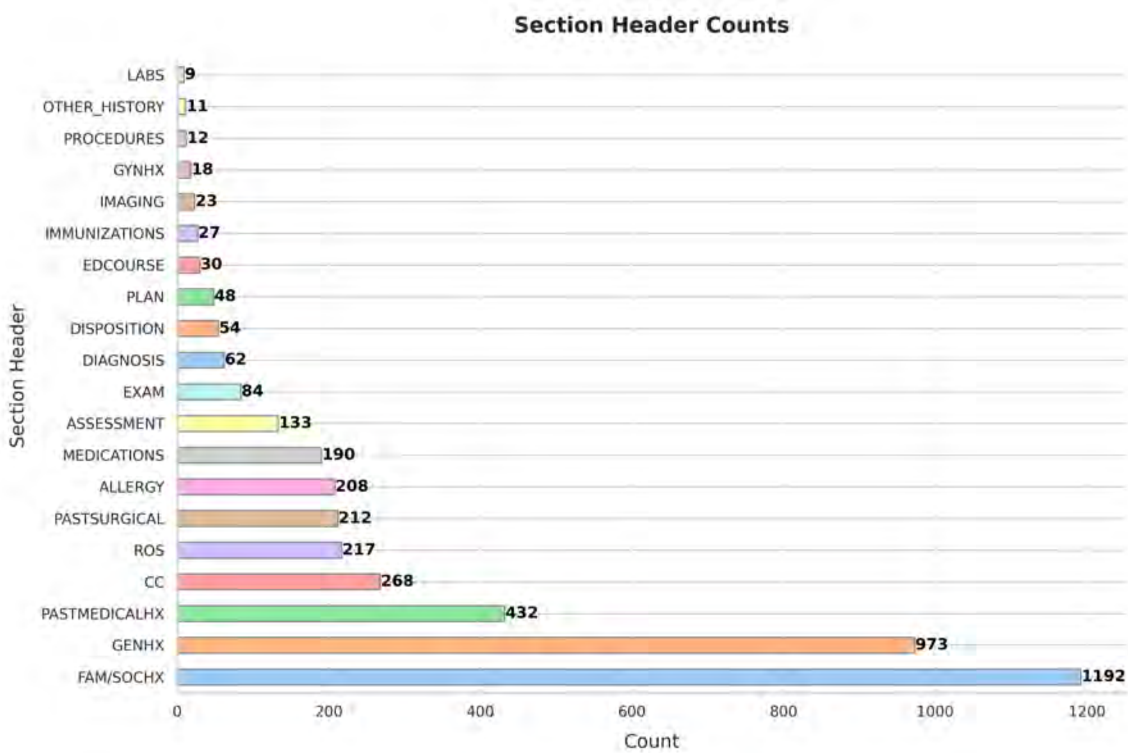

Figure 3.3: Section Header distribution of Augmented MTS-Dialog dataset

Back-translation augmentation stands out as a valuable method. It involves translating the original text into a different language and then re-translating it back into the original language. This process introduces natural linguistic variations while preserving the core meaning, thus expanding the training dataset and helping models generalize better to unseen data. To reduce translation errors, d French and Spanish were selected for their lexical proximity with English, and high-performing translation models [21]. It was implemented using the following three steps:

- **Translation:** The original text is translated from its source language (English) to a target language (French and Spanish) using a machine translation model.
- **Back-translation:** The translated text is then translated back to the original language (English) using another machine translation model.
- **Augmentation:** The back-translated text is added to the original training dataset, creating a larger and more diverse corpus.

The augmented dataset consists of 3.6k pairs of medical conversations and associated summaries created from the original 1.2k training pairs via back-translation using two languages French and Spanish, as described in the paper [28]. The distribution of section headers of augmented data is shown in figure 3.3. Back-translation can significantly increase the size of a training dataset. theoretically, it can improve the performance of the summarization model. By exposing models to different linguistic variations, back-translation can help them generalize unseen data better. Nevertheless, by increasing the diversity of training data, back-translation helps prevent overfitting, which is a common problem in NLP. The full list of normalized section headers and their counts in the augmented dataset are shown in table 3.3

Section Headers	**Counts**
FAM/SOCHX [FAMILY HISTORY/SOCIAL HISTORY]	1192
GENHX [HISTORY of PRESENT ILLNESS]	973
PASTMEDICALHX [PAST MEDICAL HISTORY]	432
CC [CHIEF COMPLAINT]	268
PASTSURGICAL [PAST SURGICAL HISTORY]	212
ALLERGY	208
ROS [REVIEW OF SYSTEMS]	217
MEDICATIONS	190
ASSESSMENT	133
EXAM	84
DIAGNOSIS	62
DISPOSITION	54
PLAN	48
EDCOURSE [EMERGENCY DEPARTMENT COURSE]	30
IMMUNIZATIONS	27
IMAGING	23
GYNHX [GYNECOLOGIC HISTORY]	18
PROCEDURES	12
OTHER_HISTORY	11
LABS	9

Table 3.3: Section Headers in the augmented Dataset.

Chapter 4

Fine-Tuning Techniques

4.1 PEFT with LoRA

In most cases, GPU hardly has enough memory to fine-tune any decoder-only LLM. To overcome this problem according to the research paper [11], we need to use Parameter Efficient Fine-tuning (PEFT).
Fine-tuning involves copying the weights from a pre-trained network and tuning them on the downstream task. This means there is a new set of weights for each task. Multi-task learning requires simultaneous access to all tasks and this is quite memory extensive. Adapters yield parameter-efficient tuning for NLP. It permits training on tasks sequentially. Tuning with adapter modules involves adding a small number of new parameters to a model, which are trained on the downstream tasks. In adapter-tuning, the parameters of the original network are frozen and therefore may be shared by many tasks. The basic architecture of this procedure is shown in Figure 4.1.

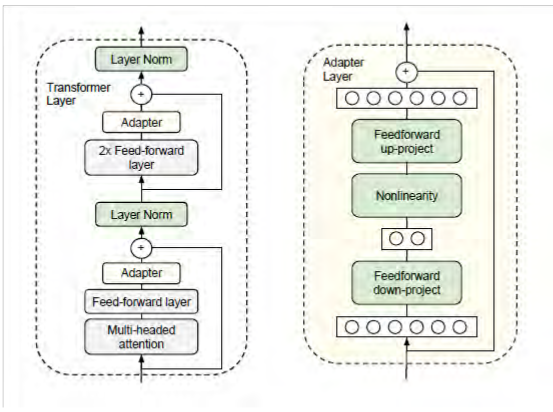

Figure 4.1: Architecture of transformer for adapter tuning [11]

One popular method of PEFT is Low-Rank Adaptation (LoRA). This technique freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the transformer architecture. The model is first initialized during full fine-tuning using pre-trained weights Φ. It is then iteratively updated to $\Phi + \Delta\Phi$ by following the gradient to maximize the conditional language modeling objective. One of the primary drawbacks of full fine-tuning is that each downstream task requires learning a different set of parameters $\Delta\Phi$. LoRA adopts a more parameter-efficient approach, where the task-specific parameter increment $\Delta\Phi = \Delta\Phi(\Theta)$ is further encoded by a much smaller-sized set of parameters Θ. LoRA proposes to use a Low-Rank representation to encode $\Delta\Phi$. LoRA uses a singular value decomposition technique (SVD) to break any matrix A down [22].

4.1.1 Singular Value Decomposition (SVD)

This paper [1] introduced the concept of Singular Value Decomposition (SVD) in terms of approximating matrices using the best lower-rank approximations. Singular Value Decomposition (SVD) is a powerful mathematical technique to factorize a matrix into three simpler matrices. It is widely applied in various fields, including signal processing, data compression, and machine learning, particularly in dimensionality reduction. Given a real or complex matrix A of size $m \times n$, the Singular Value Decomposition of A is a factorization of the form:

$$A = U\Sigma V^T \tag{4.1}$$

Where:

- U is an $m \times m$ orthogonal matrix (if A is real) or unitary matrix (if A is complex)
- Σ is an $m \times n$ diagonal matrix, with non-negative real numbers on the diagonal called singular values.
- V^T (or V^H for complex matrices) is the transpose (or conjugate transpose) of $n \times n$ orthogonal matrix (if A is real) or unitary matrix (if A is complex).

The singular values in Σ represent the "strength" of each corresponding dimension of the matrix. The number of non-zero singular values indicates the rank of the matrix.

Key Components of SVD:

1. **Left Singular Vectors (Columns of U):** These vectors span the space corresponding to the rows of A. They represent the directions in which the data associated with the rows of A has the most variance.
2. **Singular Values (Diagonal Elements of Σ):** These values measure the magnitude of the variance in the corresponding directions defined by the singular vectors. Larger singular values represent more significant dimensions in the data.
3. **Right Singular Vectors (Columns of V):** These vectors span the space corresponding to the columns of A. They represent the directions in which the data associated with the columns of A varies.

Working mechanism of SVD

- D**imensionality Reduction:** In many practical applications, the singular values tend to decrease rapidly, meaning that after the first few singular values, the remaining ones become very small. This allows one to approximate the original matrix A by keeping only the largest singular values and the corresponding singular vectors, thereby reducing the dimensionality of the data while retaining most of the important information.

$$A \approx U_k \Sigma_k V_k^T \tag{4.2}$$

  Where U_k, Σ_k, V_k^T contain only the first k singular values and their corresponding singular vectors. This is especially useful in applications like image compression, where we can represent an image with fewer dimensions while preserving its essential structure.

- **Data Compression:** SVD is used in image and data compression because it provides an efficient way to approximate a matrix with fewer parameters. In an image, for example, the matrix representing pixel values can be decomposed, and only the most significant singular values and vectors are kept, reducing the size of the data.

4.2 QLoRA

QLoRA was first proposed in the research paper [45].QLoRA stands for Quantization and Low-Rank Adapters. In this method, the original pre-trained weights of the model are quantized to 4-bit and kept fixed during fine-tuning. Then, a small number of trainable parameters in the form of lor-rank adapters are introduced during fine-tuning. These adapters are trained to adapt the pre-trained model to the specific task it is being fine-tuned for, in 32-bit floating point format. Regarding computations (like forward and backward passes during training, or inference), the 4-bit quantized weights are dequantized back to 32-bit floating-point numbers. After the fine-tuning process, the model consists of the original weights in 4-bit form, and the additional low-rank adapters in their higher precision format. The additional low-rank adapters in the QLoRA method are in a higher precision format, typically 32-bit floating-point for a few reasons:

- Higher precision allows the model to capture more subtle patterns in the data. This is particularly important for the low-rank adapters, as they are responsible for adapting the pre-trained model to the specific task it is being fine-tuned for.

- Training neural networks involves a lot of incremental updates to the weights. Weights in a higher precision format ensure that updates are accurately captured.

While quantizing all weights can save memory, the computational efficiency might not always improve. GPUs are optimized for 32-bit or bfloat16 operations. Computations in 32-bit floating-point can be faster than with lower precision. QLoRA backpropagates gradients through a frozen, 4-bit quantized pre-trained language model into Low-Rank Adapters (LoRA).

4.2.1 LoRA + int8 quantization

It is possible to combine low-rank adaptation with int8 quantization to further optimize memory usage and speed up inference on hardware with specialized instructions for int8 operations, such as modern CPUs and AI accelerators. By reducing the precision of weights and activations to 8-bit integers, it is possible to further reduce the memory footprint and increase the efficiency of operations while maintaining high accuracy on many tasks. It is quite useful for deploying LLMs to edge devices. Quantization can affect the accuracy of the model, as reducing the precision of the weights and activations can lead to loss of information.

Chapter 5

Research Methodology

5.1 Task description

This research introduces a new task in text generation. To solve this, a text-generation model must be created to generate clinical notes from the conversation between doctor and patient. A training-inference diagram of this text generation model is shown in figure 5.1.

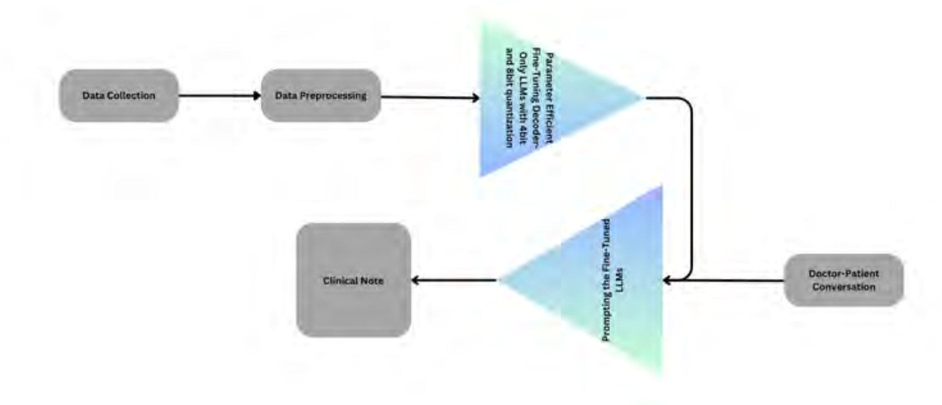

Figure 5.1: Training and Inference diagram of the clinical note generation model.

5.1.1 Training Procedure

- In Given Dataset: C, N:
  - Where $C = \{C_0, C_1, ...C_i\}$ set of doctor-patient conversations
  - AND $N = \{N_0, N_1, ...N_i\}$ set of clinical Notes
- Here, A text generative model $F(C_j)$ was proposed to develop that can generate a valid N_j which is not generated from $C_0, C_1, ...C_i$ and not in $N_0, N_1, ...N_i$

- Characteristics of a valid generated Note N_k:
  - N_k must follow the syntax of a target language.
  - N_k must maintain clinical integrity
  - N_k must follow the semantics of a target language.
  - N_k must not be a hallucination

5.1.2 Inference

- Input: C_j = Conversation between doctor and patient
- Output: N_j = Clinical Notes for C_j

5.2 Data Pre-processing

5.2.1 Removing Unnecessary Spaces

Some unnecessary tags in the dataset were removed by empty string. Certain spaces and line gaps were eliminated for text processing and analysis to streamline the training.

5.2.2 Tokenization

Tokenization is one of the most vital steps in this research. In this particular research, mostly HuggingFace models were used. For this reason, HuggingFace's AutoTokenizer class is used for Tokenization.

5.3 Training Setup

We used the online platform Kaggle to fine-tune our dataset, the hardware configuration is shown in the table 5.1.

Component	Kaggle's Provided setup
CPU	Intel(R) Xeon(R) CPU @ 2.00GHz
GPU	Tesla P100-PCIE-16GB
CUDA Version	12.4
VRAM	16 GB
Available RAM	29 GB

Table 5.1: Hardware setup for training LLMs

5.4 Fine-Tuning the Llama and Mistral variants

At first, the author of this paper tried different sequence-to-sequence models with data augmentation techniques to beat the current SOTA model but the result was not satisfactory. Next, several state-of-the-art decoder-only models, such as variants of Mistral and Llama, were evaluated. The "Meta-Llama-3-8B" model, an updated version of the Llama family with 8B parameters, outperformed the state-of-the-art Flan-T5-large model in the Rouge and BERT metrics while the "Mistral-7B-v0.3" outperformed in the BERT metric only. A smaller variant of the Llama family "Llama-3.2-3B" was also fine-tuned. Due to low hardware resources full model fine-tuning was not feasible. The decoder-only models were fine-tuned using parameter-efficient fine-tuning (PEFT) [11] with the low-rank adaptation (LoRA) [22] method. Both 8-bit and 4-bit quantization were used while loading the models. This method is known as QLORA which stands for Quantization and Low-Rank-Adapters [45]. In this method, the original pre-trained weights of the model are quantized to 8-bit or 4-bit and kept fixed during fine-tuning. Then, a few trainable parameters in the form of low-rank adapters are introduced during fine-tuning [22]. These adapters are trained to adapt the pre-trained model to the specific task it is being fine-tuned for, in the 32-bit floating-point format. Regarding computations (like forward and backward passes during training, or inference), the 8-bit or 4-bit quantized weights are dequantized back to 32-bit floating-point numbers. After the fine-tuning process, the model consists of the original weights in 8-bit or 4-bit form, and the additional low-rank adapters in their higher precision format.

A significant percentage of this research and analysis is devoted to fine-tuning. For this procedure, the **MTS-DIALOG** dataset was used. A variant of the Llama model, **"Meta-Llama-3-8B"** and **"Llama-3.2-3B"** was used for fine-tuning with the following hyper-parameters. For Mistral, the variant was **"Mistral-7B-v0.3"**. The model was loaded with 8-bit quantization for Llama and 4-bit quantization for Mistral and the following LoRA configurations were maintained and described in the table 5.2.

LoRA Configuration	**Value**
lora_alpha	16
lora_dropout	0.1
LoRA attention dimension (rank), r	64
target_modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj, lm_head
bias	none
task_type	CASUAL_LM

Table 5.2: LoRA Configuration

The prompt that was used as a prefix to summarize the doctor-patient dialogue to generate clinical notes was: *"Summarize the following patient-doctor dialogue. Include all medically relevant information, including family history, diagnosis, past medical (and surgical) history, immunizations, lab results, and known allergies."* A sample training and validation prompt is shown in the Figure 5.2 and Figure 5.3 respectively.
While training, the following "TrainingArguments" class's configuration was used and it is shown in the table 5.3.
It is important to keep in mind that to improve performance, several experiments and parameter sweeps were conducted before selecting these hyperparameters.

Hyper-parameter	Value
per_device_train_batch_size	1
logging_steps	100
warmup_steps	0.03
save_strategy	epoch
group_by_length	True
lr_scheduler_type	constant
max_seq_length	512

Table 5.3: TrainingArguments Configuration

```
<|begin_of_text|>### Instruction:
Summarize the following patient-doctor dialogue. Include all medically relevant information, including
family history, diagnosis, past medical (and surgical) history, immunizations, lab results and known a
llergies.

### Input:
Doctor: What brings you back into the clinic today, miss?
Patient: I came in for a refill of my blood pressure medicine.
Doctor: It looks like Doctor Kumar followed up with you last time regarding your hypertension, osteoar
thritis, osteoporosis, hypothyroidism, allergic rhinitis and kidney stones.  Have you noticed any chan
ges or do you have any concerns regarding these issues?
Patient: No.
Doctor: Have you had any fever or chills, cough, congestion, nausea, vomiting, chest pain, chest press
ure?
Patient: No.
Doctor: Great. Also, for our records, how old are you and what race do you identify yourself as?
Patient: I am seventy six years old and identify as a white female.

### Response:
The patient is a 76-year-old white female who presents to the clinic today originally for hypertension
and a med check.  She has a history of hypertension, osteoarthritis, osteoporosis, hypothyroidism, all
ergic rhinitis and kidney stones.  Since her last visit she has been followed by Dr. Kumar.  Those iss
ues are stable.  She has had no fever or chills, cough, congestion, nausea, vomiting, chest pain, ches
t pressure.<|end_of_text|>
```

Figure 5.2: Training-Prompt

```
### Instruction:
Summarize the following patient-doctor dialogue. Include all medically relevant information, including family histo
ry, diagnosis, past medical (and surgical) history, immunizations, lab results and known allergies.

### Input:
Doctor: When did your pain begin?
Patient: I've had low back pain for about eight years now.
Doctor: Is there any injury?
 Patient: Yeah, it started when I fell in an A B C store.
Doctor: How old are you now?
Patient: I'm twenty six.
Doctor: What kind of treatments have you had for this low back pain?
Patient: Yeah, I got referred to P T, and I went, but only once or twice, um, and if I remember right, they only di
d the electrical stimulation, and heat.
Doctor: I see, how has your pain progressed over the last eight years?
Patient: It's been pretty continuous, but it's been at varying degrees, sometimes are better than others.
Doctor: Do you have any children?
Patient: Yes, I had my son in August of two thousand eight, and I've had back pain since giving birth.
Doctor: Have you had any falls since the initial one?
Patient: Yes, I fell four or five days ago while I was mopping the floor.
Doctor: Did you land on your lower back again?
Patient: Yes, right onto my tailbone.
Doctor: Did that make the low back pain worse?
Patient: Yes.
Doctor: Have you seen any other doctors for this issue?
Patient: Yes, I saw Doctor X on January tenth two thousand nine, and I have a follow up appointment scheduled for F
ebruary tenth two thousand nine.

### Response:
```

Figure 5.3: Validation-Prompt

Pytorch "generate()" method is used for generating clinical notes from the finetuned LLMs. The hyper-parameters used for generating clinical notes are:

- max_new_tokens = 512
- do_sample = True
- temperature=0.8
- pad_token_id = tokenizer.eos_token_id

5.5 Evaluation Metrics

In this research ROUGE and BERTScore evaluation metrics are used to evaluate the model's generated clinical notes.

5.5.1 ROUGE Evaluation Metric

Chin-Yew Lin introduced ROUGE in 2004 [2]. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics used for evaluating automatic text summarization and machine translation by comparing the overlap between the machine-generated text (candidate summary) and a human reference summary. The ROUGE family includes several variations, such as **ROUGE-N**, **ROUGE-L**, and **ROUGE-W**. Each measures different types of overlap.

ROUGE-N (N-gram overlap)

ROUGE-N measures the overlap of N-grams between the candidate summary and the reference summary. An N-gram is a contiguous sequence of N items from a given text. Common values of N include 1 (unigrams) and 2 (bigrams). Mathematical Expression for ROUGE-N:

$$\text{ROUGE-N} = \frac{\sum_{\text{n-gram}\in\text{reference}} \min(\text{Count}_{\text{ref}}(\text{n-gram}), \text{Count}_{\text{cand}}(\text{n-gram}))}{\sum_{\text{n-gram}\in\text{reference}} \text{Count}_{\text{ref}}(\text{n-gram})} \tag{5.1}$$

Where:

- $Count_{ref}(n-gram)$ is the number of times the n-gram appears in the reference summary.
- $Count_{cand}(n-gram)$ is the number of times the n-gram appears in the candidate summary.
- The numerator is the count of overlapping n-grams, while the denominator is the total number of n-grams in the reference summary.

For example, ROUGE-1 measures the overlap of unigrams, while ROUGE-2 measures the overlap of bigrams.

ROUGE-L (Longest Common Subsequence)

ROUGE-L evaluates the longest common subsequence (LCS) between the candidate and reference summaries. LCS is a measure of the longest-ordered sequence of words that appear in both summaries, though they may not be consecutive. Mathematical Expression for ROUGE-L:

$$\text{ROUGE-L} = F_\beta = \frac{(1+\beta^2)\cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{5.2}$$

Where:

- $Precision = \frac{LCS(C,R)}{|C|}$
- $Recall = \frac{LCS(C,R)}{|R|}$
- $LCS(C,R)$ is the length of the longest common subsequence between the candidate summary C and reference summary R.
- $|C|$ and $|R|$ are the lengths of the candidate and reference summaries, respectively.
- β is usually set to 1 to equally balance precision and recall.

5.5.2 BERTScore

BERTScore is an advanced metric for evaluating text generation tasks (such as machine translation, summarization, and paraphrasing). It leverages deep contextual embeddings from pre-trained models like BERT (Bidirectional Encoder Representations from Transformers) to assess the semantic similarity between a candidate text and a reference text. The method was introduced in this paper [16]. Unlike traditional n-gram-based metrics (such as BLEU, and ROUGE), BERTScore captures nuanced contextual meaning by comparing the embeddings of tokens (words) in the candidate and reference sentences. BERTScore computes three core components: Precision, Recall, and F1-score. It computes these values by comparing the similarity of token embeddings between the candidate and reference texts.

Embedding-based Token Similarity

If $C = [c_1, c_2, \ldots, c_n]$ and $R = [r_1, r_2, \ldots, r_m]$ are the candidate and reference tokenized sequences, respectively. Each token c_i and r_j is embedded into a dense vector using a pre-trained model such as BERT. Let c_i and r_j represent the corresponding embeddings of the tokens c_i and r_j. For each token in the candidate sentence, BERTScore computes its similarity to each token in the reference sentence using a cosine similarity function.

$$\text{Sim}(c_i, r_j) = \frac{\mathbf{c}_i \cdot \mathbf{r}_j}{\|\mathbf{c}_i\|\|\mathbf{r}_j\|} \tag{5.3}$$

Where:

- $c_i \cdot r_j$ is the dot product of the token embeddings.
- $\|\mathbf{c}_i\|$ and $\|\mathbf{r}_j\|$ are the magnitudes of the embeddings.

Precision

Precision in BERTScore measures how much of the candidate text is semantically similar to the reference text. For each token in the candidate text C, we find the most similar token in the reference text R, based on cosine similarity shown in equation 5.3.

$$\text{Precision} = \frac{1}{n} \sum_{i=1}^{n} \max_{j} \text{Sim}(c_i, r_j) \tag{5.4}$$

Where:

- $\max_j \text{Sim}(c_i, r_j)$ finds the maximum similarity between the candidate token c_i and all reference tokens r_j

Recall

Recall measures how much of the reference text is captured by the candidate text. For each token in the reference text R, the most similar token in the candidate text C can be found, based on cosine similarity.

$$\text{Recall} = \frac{1}{m} \sum_{j=1}^{m} \max_{i} \text{Sim}(r_j, c_i) \tag{5.5}$$

Where:

- $\max_i \text{Sim}(r_j, c_i)$ finds the maximum similarity between the reference token r_j and all candidate tokens c_i.

F1-score

BERTScore computes an F1-score as the harmonic mean of Precision and Recall, giving a balanced evaluation of both:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5.6}$$

Final BERTScore Calculation

To produce the final BERTScore for a candidate-reference pair, Precision, Recall, and F1-score are calculated based on the cosine similarities of the token embeddings. Typically, the F1 score is used as the primary metric to balance the trade-off between precision and recall.

Chapter 6

Results

Method	Rouge1	Rouge2	RougeL	BERTScore_F1
Llama-3.2-3B	0.3686	0.1517	0.2895	0.8901
Llama-3-8B	**0.4574**	0.2079	0.3636	**0.9060**
Llama-3-8B + Data Augmentation	0.4131	0.1888	0.3410	0.8956
Mistral-7B	0.4211	0.1622	0.3218	0.8985
Mistral-7B + Data Augmentation	0.3571	0.1599	0.2969	0.8869

Table 6.1: Results of the decoder-only llms

The results are shown after fine-tuning the Llama3-8B and Mistral-7B models in Table 6.1. Rouge and BERTScore were used as evaluation metrics. The Performances with and without data augmentation are displayed in the table. The fine-tuned Llama-3-8B model achieved the highest performance in the Rouge1 score in the summarization task. It achieved Rouge1 score of 0.4574, Rouge2 score of 0.2079, RougeL score of 0.3636 and BERTScore_F1 of 0.9060. For some reason, Llama-3-8B with Data Augmentation performs lower than this. With data augmentation, we get a Rouge1 score of 0.4131, a Rouge2 score of 0.1888, a RougeL score of 0.3410, and BERTScore_F1 score of 0.8956. A smaller version of Llama model, "Llama-2ż-3B" is also fine-tuned but the performance was not good enough. The model gave Rouge1 score of 0.3686 and BERTScore_F1 of 0.8901. The Finetuned Mistral model was also giving SOTA BERTScore_F1, 0.8985. But in the case of Rouge scores, it is lagging behind Llama3. With data augmentation, the performance of the Mistral-7B model decreased significantly. The mistral-7 B model was fine-tuned with 4bit quantization whereas the variants Llama model were fine-tuned with 8bit quantization. This could be a reason that the mistral-7B is not performing as well as Llama-3-8B. Data augmentation did not improve the performance of either of the models. Both Mistral and Llama model variants are performing better without data augmentation. A similar type of situation is explained in the research paper [23].

There was an annotation error in the validation dataset shown in Figure 6.1. The results indicate that the model's generated summary aligns with the context of the dialogue, whereas the annotated summary contradicts it. This demonstrates the model's ability to handle critical situations effectively. Here the Llama model-generated output is more aligned with the core concept of the dialogue than the Mistral model.

Dialogue	Annotated Note	Llama Model Generated Note	Mistral Model Generated Note
Doctor: Have you been experiencing any mental difficulties or confusion? **Patient**: No. **Doctor**: Any hallucinations? Are you seeing hearing thing that is not real? **Patient**: No	Confusion and hallucinations.	No headaches. No visual, hearing, or swallowing difficulties.	No headaches. No visual, hearing, or swallowing difficulties. No changes in bowel or urinary habits. No chest pain or shortness of breath.

Figure 6.1: Annotation-Error

Besides these evaluation metrics, Our fine-tuned Mistral and Llama models did not show any hallucination while generating new Clinical Notes from the doctor-patient conversation. Some samples of different model-generated notes and the reference notes are shown in the following tables 6.2, 6.3, and 6.4.

Generated Note	**Reference Note**
The patient is a 57-year-old female who has a history of bilateral bunionectomy performed approximately 13 years ago and is now with severe bilateral bunion deformities, worse on the right than the left. She has also noted some pain in the right second metatarsophalangeal joint as well as the right second toe deformity. She has some bilateral second and third toe deformities consistent with arthritis. The patient works in a hospital where she is on her feet all day long. She does use orthotics, but still has significant bunion pain.	The patient is a 57-year-old female being seen today for evaluation of pain and symptoms related to a recurrent bunion deformity in bilateral feet, right greater than left. The patient states she is having increasing symptoms of pain and discomfort associated with recurrence of bunion deformity on the right foot and pain localized to the second toe and MTP joint of the right foot as well. The patient had prior surgery performed approximately 13 years ago. She states that since the time of the original surgery the deformity has slowly recurred, and she has noticed progressive deformity in the lesser toes at the second and third toes of the left foot and involving the second toe of the right foot. The patient is employed on her feet as a hospital employee and states that she does wear a functional orthotic which does provide some relief of forefoot pain although not complete.
The patient is on methadone maintenance treatment (MMT), inhalation drugs IV drug abuse. He lives with a room-mate. He is being followed for depression issues.	The patient has a history of inhalation on intravenous drug abuse. She is currently on methadone maintenance. She is being followed in Psychiatry for depression and substance abuse issues. She lives with a room-mate.
Back pain after a fall.	Back pain after a fall.
He is a nonsmoker. He works as a data entry operator for an information technology company.	He denies tobacco or alcohol. He has what sounds like a data entry computer job.

Table 6.2: Llama-3-8B model's generated clinical notes sample

Generated Note	Reference Note
This 57-year-old female is status post bunionectomy 13 years ago for the removal of bunions. She is here today to discuss the fact that she is developing bunions on her feet again, she has discomfort over the dorsal aspect of her feet, and she has also developed arthritis in the second and third toes on the left side. She states that she works at the hospital and she is on her feet all day. She does wear orthotics for shoe gear. She states that the orthotics do provide her with some relief, but she is still having significant pain. She desires treatment at this time.	The patient is a 57-year-old female being seen today for evaluation of pain and symptoms related to a recurrent bunion deformity in bilateral feet, right greater than left. The patient states she is having increasing symptoms of pain and discomfort associated with recurrence of bunion deformity on the right foot and pain localized to the second toe and MTP joint of the right foot as well. The patient had prior surgery performed approximately 13 years ago. She states that since the time of the original surgery the deformity has slowly recurred, and she has noticed progressive deformity in the lesser toes at the second and third toes of the left foot and involving the second toe of the right foot. The patient is employed on her feet as a hospital employee and states that she does wear a functional orthotic which does provide some relief of forefoot pain although not complete.
The patient is currently following up with Dr. ABC for depression issues and is on methadone maintenance. The patient has a room-mate.	The patient has a history of inhalation on intravenous drug abuse. She is currently on methadone maintenance. She is being followed in Psychiatry for depression and substance abuse issues. She lives with a room-mate.
Back pain after a fall.	Back pain after a fall.
Operates data for an IT company. Denies ETOH/Tobacco/Illicit drug use.	He denies tobacco or alcohol. He has what sounds like a data entry computer job.

Table 6.3: Mistral-7B model's generated clinical notes sample

Generated Note	Reference Note
The patient is a 57-year-old female who has had multiple procedures on her feet to remove bunions, but unfortunately, the procedure appears to have been unsuccessful. She states that the feet have been mostly normal for about 13 years, but within the last few months she has noted some redness and swelling of her feet. She states that she has been wearing some orthotics as well as some foot gear to help her feet, but she still has significant amount of redness and swelling and some pain. She states that the redness and swelling is much worse on her right foot versus her left foot. When I asked her if the pain was getting worse, she states that it is and she is quite uncomfortable. When I examined her feet, I noted some significant deformity of the second toe and MTP joint as well as some very apparent redness and swelling. I also noted some mild amount of arthritis in the second and third toes of her left foot. I questioned her further as to the diagnosis of erythema nodosum, which she states she is not familiar with, but she does note some redness and swelling in her feet. I also questioned her further as to what profession she is in and she states that she works as a janitor in a hospital, meaning she is on her feet all day long. She states that the foot gear and orthotics she is currently wearing gives her some relief, but she is still in significant amount of pain.	The patient is a 57-year-old female being seen today for evaluation of pain and symptoms related to a recurrent bunion deformity in bilateral feet, right greater than left. The patient states she is having increasing symptoms of pain and discomfort associated with recurrence of bunion deformity on the right foot and pain localized to the second toe and MTP joint of the right foot as well. The patient had prior surgery performed approximately 13 years ago. She states that since the time of the original surgery the deformity has slowly recurred, and she has noticed progressive deformity in the lesser toes at the second and third toes of the left foot and involving the second toe of the right foot. The patient is employed on her feet as a hospital employee and states that she does wear a functional orthotic which does provide some relief of forefoot pain although not complete.
The patient lives with a roommate and is on methadone maintenance treatment for heroin addiction and depression issues.	The patient has a history of inhalation on intravenous drug abuse. She is currently on methadone maintenance. She is being followed in Psychiatry for depression and substance abuse issues. She lives with a room-mate.
Back pain.	Back pain after a fall.
He is a nonsmoker and nondrinker and he works as a data operator for an IT company.	He denies tobacco or alcohol. He has what sounds like a data entry computer job.

Table 6.4: Llama-3-3B model's generated clinical notes sample

Chapter 7

Limitations

7.1 Training and Inference time

From table 7.1 it is visible that the inference time for Mistral-7B is better than Llama-3-8B. However, for real-world applications, it needs to be improved more. Training time for both the models is quite long also 7.1. In case of emergency, this lengthiness could cause harm to the patient.

Model	**Training Time** (sec/epoch)
Llama-3-3B	1406.6
Llama-3-8B	2588.9
Llama3-8B + Data Augmentation	6732.2
Mistral-7B	2856.4
Mistral-7B + Data Augmentation	7374.7

Table 7.1: Training time per epoch

Model	**Average Inference Time** (per note generation)
Llama-3-3B	12.49 (On Tesla T4 GPU)
Llama-3-8B	13.41 (On Tesla T4 GPU)
Mistral-7B	8.30 (On Tesla T4 GPU)

Table 7.2: Time needed per note generation

7.2 Hardware Limitation

Hardware limitation is one of the fundamental problems in this work. For this research, P100 GPU is used on the online platform Kaggle. But it has only 16GB of VRAM which is not sufficient to train other variants of the Llama and mistral models like Llama3-70B. It is clear from the results table 6.1 that the larger variants of the decoder-only LLMs are giving better results than the smaller variants. However, due to Hardware limitations, it is not feasible to fine-tune those models on the Kaggle platform. For this reason, the author in this research has not tried larger variants of Llama and Mistral. Since 16GB VRAM was not sufficient, the hyper-parameter "per_device_train_batch_size" was kept to a minimum throughout the training. For this reason, it took too much time to train any model 7.1.

7.3 Gender Bias

All the fine-tuned models had some gender biases. Since the "MTS-DIALOG" dataset is a short doctor-patient conversation dataset, sometimes it is quite difficult to understand the patient's gender from the conversation. In these situations, all of the fine-tuned models mostly assume the patient's gender as male. In a few cases, the model predicts the gender as female while in the annotation the patient is identified as a male. In some cases in the dataset, the gender pronoun is used incorrectly. The author of this paper assumes that most of the model's pre-training dataset could be biased, which is causing this type of problem. In some cases in the dataset, the gender pronoun is used incorrectly also. From table 7.3 we can get a more clear idea about it. In the first example, the model treated the patient as male because in the conversation the Doctor addressed the patient as male. However, in the Reference clinical note, the patient is addressed as female, which is wrong. So here the model predicted the gender correctly. The second example in the table shows that it is quite impossible to predict the gender of the patient from the conversation. However, in the reference note, the patient is labeled as male. In the generated note, the patient is addressed as Female by the model. The researcher of this paper hypothesizes that the patient's non-smoking and non-drinking status influenced the model's classification of the patient as female. Another similar example is the third row in the table. In the fourth example of the table, it is not possible to get a clear idea about the patient's gender from the conversation but the model addressed the patient as male in this case. To overcome this issue the prompt is updated for inferencing. The prompt that was used as a prefix to generate clinical notes and identify the patient's gender properly is:

"Summarize the following patient-doctor dialogue. To ensure a comprehensive summary, follow these steps:

1. *Gender Identification: Identify the patient's gender based on the context and use appropriate pronouns throughout the summary.*
2. *Medical History: Summarize the patient's family history, past medical and surgical history, and known allergies. Ensure each detail is clearly categorized.*
3. *Current Visit: Identify the main concerns discussed, including symptoms, lab results, and diagnosis.*
4. *Immunizations: List any relevant immunization history.*
5. *Plan and Recommendations: Note any treatment plan, medications prescribed, or follow-up actions recommended by the doctor. Ensure your summary flows logically, preserving the order of the conversation, while focusing on medically relevant details."*

However, even after using this prompt, there was still no significant improvement happened.

No.	Conversation	Generated Note	Reference Note
1	Doctor: When are you planning to quit your I V and inhalation drugs Mister X Y Z? Patient: I am on treatment now. They are following me up for depression issues too. Doctor: Okay yes, you are on methadone maintenance. Patient: Yes. Doctor: Who lives with you? Patient: I have a roommate.	He is on methadone maintenance treatment and is being followed by psychiatry for depression issues. He lives with a room-mate.	The patient has a history of inhalation on intravenous drug abuse. She is currently on methadone maintenance. She is being followed in Psychiatry for depression and substance abuse issues. She lives with a room-mate.
2	Doctor: So, where do you work? Patient: Oh, I am a data operator for an I T company. Doctor: Okay. Doctor: Do you smoke or drink? Patient: No, I don't!	She is a nonsmoker and nondrinker. She works as a data operator for an IT company.	He denies tobacco or alcohol. He has what sounds like a data entry computer job.
3	Doctor: Do you smoke? Patient: No, I do not. Doctor: How about alcohol. Patient: No, I don't drink alcohol as well. Doctor: Okay let me ask you, do you have any children or are you married? Patient: No, I am single, thus no kids. Doctor: Okay, where do you work? Patient: I work as a payroll representative. I used to do a lot of work in jewelry business, with chemical. Doctor: interesting. Patient: Yeah.	No smoking or drinking. She is single with no children. She works as a payroll representative. She has a history of working with chemicals in the jewelry business.	He is a non-cigarette smoker and non-ETOH user. He is single and he has no children. He works as a payroll representative and previously did lot of work in jewelry business, working he states with chemical.
4	Doctor: Do you drink or smoke? Or take any other kind of drugs? Patient: I used to smoke and drink, but I quit years ago. Maybe it was like in ninety two. Doctor: How many cigarettes were you smoking then? Patient: You see that is a tough one to remember. It was anywhere around thirty packs per year.	He has a thirty pack year smoking history, but quit in 1992. He has a history of alcohol abuse, but quit in 1992 as well. He denies any current drug use.	ETOH abuse (quit '92), 30pk-yr Cigarettes (quit '92)

Table 7.3: Gender Bias in the fine-tuned models

Chapter 8

Future Work

8.1 Development of new Medical Corpus

The amount of data in the medical domain is very limited. It is also very hard to get access to this type of data because of the Physician-patient privilege. Most of the patients are not comfortable sharing their private data. It is essential to develop more data which will make it easier to create an automatic clinical note generation system. The dataset used in this research is short conversations between doctors and patients. More and longer real-world doctor-patient conversation corpus is needed in the future to improve the quality of clinical note generation.

8.2 Development of Pre-trained Medical LLMs

Domain-specific pre-trained decoder-only LLMs have improved the domain-specific task a lot in recent years. A pre-trained model like Code Llama is one example in the coding domain [35]. Developing a decoder-only model that is pre-trained on a large medical corpus might help to create clinical note-generation tasks.

8.3 Cross-lingual Clinical Notes

In diverse healthcare settings, doctors and patients may converse in different languages. A cross-lingual summary can help bridge this language gap by automatically summarizing conversations in one language and translating the summary into another, making it accessible to a wider range of healthcare providers. Creating a cross-lingual clinical note generator could greatly impact this domain.

8.4 Speech-to-Note generator

In this research, text data from doctor-patient conversations is used. Generating clinical notes directly from the spoken interactions between doctors and patients could provide a more accurate and practical solution.

8.5 Calculating Hallucination in LLM-Generated Notes

In large language models (LLMs), "hallucination" refers to generating text that is factually incorrect or unsupported by the input data. Managing hallucination is especially critical in clinical note generation, where accuracy is paramount. Various methods can assess hallucination in LLM-generated clinical notes, with manual evaluation being the most precise. In this approach, human experts compare the generated notes against the source text to pinpoint factual inconsistencies. This process helps to identify and address hallucinations, thereby enhancing the reliability of LLM-generated clinical notes.

8.6 Gender Bias reduction during fine-tuning

The gender bias problem was discussed in 7.3. It is also shown that just updating the prompt while inferencing is not a solution to this problem. The hypothesis of the author of this paper suggests these steps to overcome the problem. The wrongly addressed gender in the dataset should be corrected manually in the reference note and the patient's gender information should be included in the conversation. After that, the prompt should be updated by including the information to detect the correct gender of the patient and use this prompt to fine-tune the model. By doing all these tasks it may be possible to solve the problem.

Chapter 9

Conclusion

This research involved experimenting with various decoder-only transformer architectures to fine-tune models for generating clinical notes by summarizing conversations between doctors and patients. The results demonstrate that decoder-only models like Llama3 and Mistral outperform classical encoder-decoder models like Flan-T5 and Pegasus in summarizing medical discussions. Larger models give better results than smaller ones. A state-of-the-art (SOTA) Rouge1 score was achieved by fine-tuning the Llama-3-8B model. State-of-the-art (SOTA) BERTScore_F1 is achieved by both the Llama-3-8B and Mistral-7B models. The Flan T5 Large model by the WnagLab team (2023 MEDIQA-Chat challenge) [27] Rouge scores of 0.4466 for Rouge1, 0.2282 for Rouge2, 0.3837 for RougeL, 0.7307 for BERTScore. However significant improvements were observed when we used the pre-trained decoder only **Llama-3-8B** model for fine-tuning. The Llama3-8B model achieved a Rouge1 score of **0.4574**, reflecting a **2.42%** improvement. Additionally, the BERTScore_F1 reached 90.60, demonstrating superior performance in both metrics. These improvements indicate that there is a substantial enhancement in capturing more complex sentence structures and content relationships. It was also demonstrated that the model performs correctly even when data annotations are incorrect. These results indicate that state-of-the-art performance was achieved on the MTS-DIALOG corpus.

Bibliography

[1] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, 1936.

[2] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[3] G. Hripcsak, D. K. Vawdrey, M. R. Fred, and S. B. Bostwick, "Use of electronic clinical documentation: Time spent and team interactions," *Journal of the American Medical Informatics Association*, vol. 18, no. 2, pp. 112–117, 2011.

[4] K. Cho, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[5] I. Sutskever, "Sequence to sequence learning with neural networks," *arXiv preprint arXiv:1409.3215*, 2014.

[6] H. Lin, Z. Shi, and Z. Zou, "Maritime semantic labeling of optical remote sensing images with multi-scale fully convolutional network," *Remote Sensing*, vol. 9, p. 480, May 2017. DOI: 10.3390/rs9050480.

[7] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[8] P. Liu, M. Saleh, E. Pot, *et al.*, *Generating wikipedia by summarizing long sequences. arxiv preprint arxiv: 180110198*, 2018.

[9] A. Radford, "Improving language understanding by generative pre-training," 2018.

[10] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *arXiv preprint arXiv:1803.02155*, 2018.

[11] N. Houlsby, A. Giurgiu, S. Jastrzebski, *et al.*, "Parameter-efficient transfer learning for nlp," in *International conference on machine learning*, PMLR, 2019, pp. 2790–2799.

[12] M. Lewis, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[14] N. Shazeer, "Fast transformer decoding: One write-head is all you need," *arXiv preprint arXiv:1911.02150*, 2019.

[15] B. Zhang and R. Sennrich, "Root mean square layer normalization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[17] T. B. Brown, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[18] D. M. Mann, J. Chen, R. Chunara, P. A. Testa, and O. Nov, "Covid-19 transforms health care through telemedicine: Evidence from the field," *Journal of the American Medical Informatics Association*, vol. 27, no. 7, pp. 1132–1135, 2020.

[19] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[20] N. Shazeer, "Glu variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[21] J. Tiedemann and S. Thottingal, "Opus-mt–building open translation services for the world," in *Proceedings of the 22nd annual conference of the European Association for Machine Translation*, 2020, pp. 479–480.

[22] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[23] V. Nair, N. Katariya, X. Amatriain, I. Valmianski, and A. Kannan, "Adding more data does not always help: A study in medical conversation summarization with pegasus," *arXiv preprint arXiv:2111.07564*, 2021.

[24] W. Fedus, J. Dean, and B. Zoph, "A review of sparse expert models in deep learning," *arXiv preprint arXiv:2209.01667*, 2022.

[25] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[26] B. Workshop, T. L. Scao, A. Fan, *et al.*, "Bloom: A 176b-parameter open-access multilingual language model," *arXiv preprint arXiv:2211.05100*, 2022.

[27] A. B. Abacha, W.-w. Yim, G. Adams, N. Snider, and M. Yetisgen-Yildiz, "Overview of the mediqa-chat 2023 shared tasks on the summarization & generation of doctor-patient conversations," in *Proceedings of the 5th Clinical Natural Language Processing Workshop*, 2023, pp. 503–513.

[28] A. B. Abacha, W.-w. Yim, Y. Fan, and T. Lin, "An empirical study of clinical note generation from doctor-patient encounters," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 2291–2302.

[29] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai, "Gqa: Training generalized multi-query transformer models from multi-head checkpoints," *arXiv preprint arXiv:2305.13245*, 2023.

[30] A. Chowdhery, S. Narang, J. Devlin, *et al.*, "Palm: Scaling language modeling with pathways," *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 1–113, 2023.

[31] A. Q. Jiang, A. Sablayrolles, A. Mensch, *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[32] Y. Mathur, S. Rangreji, R. Kapoor, M. Palavalli, A. Bertsch, and M. R. Gormley, "Summqa at mediqa-chat 2023: In-context learning with gpt-4 for medical summarization," *arXiv preprint arXiv:2306.17384*, 2023.

[33] K. Milintsevich and N. Agarwal, "Calvados at mediqa-chat 2023: Improving clinical note generation with multi-task instruction finetuning," in *Proceedings of the 5th Clinical Natural Language Processing Workshop*, 2023, pp. 529–535.

[34] K. B. Ozler and S. Bethard, "Clulab at mediqa-chat 2023: Summarization and classification of medical dialogues," in *Proceedings of the 5th Clinical Natural Language Processing Workshop*, 2023, pp. 144–149.

[35] B. Roziere, J. Gehring, F. Gloeckle, *et al.*, "Code llama: Open foundation models for code," *arXiv preprint arXiv:2308.12950*, 2023.

[36] A. Sharma, D. Feldman, and A. Jain, "Team cadence at mediqa-chat 2023: Generating, augmenting and summarizing clinical dialogue with large language models," in *Proceedings of the 5th Clinical Natural Language Processing Workshop*, 2023, pp. 228–235.

[37] D. Srivastava, "Iuteam1 at mediqa-chat 2023: Is simple fine tuning effective for multilayer summarization of clinical conversations?" *arXiv preprint arXiv:2306.04328*, 2023.

[38] K. Suri, S. Saha, and A. Singh, "Healthmavericks@ mediqa-chat 2023: Benchmarking different transformer based models for clinical dialogue summarization," in *Proceedings of the 5th Clinical Natural Language Processing Workshop*, 2023, pp. 472–489.

[39] X. Tang, A. Tran, J. Tan, and M. Gerstein, "Gersteinlab at mediqa-chat 2023: Clinical note summarization from doctor-patient conversations through finetuning and in-context learning," *arXiv preprint arXiv:2305.05001*, 2023.

[40] H. Touvron, T. Lavril, G. Izacard, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[41] H. Touvron, L. Martin, K. Stone, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[42] J. Wang, Z. Yao, A. Mitra, S. Osebe, Z. Yang, and H. Yu, "Umass_bionlp at mediqa-chat 2023: Can llms generate high-quality synthetic note-oriented doctor-patient conversations?" *arXiv preprint arXiv:2306.16931*, 2023.

[43] B. Zhang, R. Mishra, and D. Teodoro, "Ds4dh at mediqa-chat 2023: Leveraging svm and gpt-3 prompt engineering for medical dialogue classification and summarization," *medRxiv*, pp. 2023–06, 2023.

[44] H. W. Chung, L. Hou, S. Longpre, *et al.*, "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[45] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[46] A. Dubey, A. Jauhri, A. Pandey, *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[47] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *Neurocomputing*, vol. 568, p. 127 063, 2024.